Commodore INFO

PRIJS ƒ 7.25/Bfr. 135

EXTRA: 40 pagina's EB-Nieuws

8 710966 001332

ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS

JAARGANG 6, NO. 6, sept./okt. 1989



LISTINGS

- Super Lichtkrant C-64
- Movie C-64
- Shoot a Skeet C-64
- Tafels C-64
- UFO C-64
- Multiple Choice C-64
- Spreadsheet C-64
- Grafiek C-128
- Slang C-16
- Vijftigen C-16

Universal Scanner

I.F.F.-bestanden

Graphics op de C-64

KIWI-load

Strijd op de zeebodem

Tips en Trucs C-64

Vaste rubrieken

Geos Info

Basic Miniatuurtjes

Geos Machinetaal

Commodore Info

Verschijnt 8x per jaar
Jaarg.6, no.6, sept/okt. 1989

Uitgave:
Sala Communications

Uitgever:
Vic Sharfman

Redactie:
Ir. L. Sala hoofdredacteur
J. Bodzinga adj. hoofdred.
drs. J. Boers eindredacteur
drs. M. de Rooij &
J. Broekhuizen productie
drs. H. Zoete, H. Smeenk, drs. U. Schuurmans, R. Goudriaan, B. Munniksma, B. Venema, P. Boncz, MGCC/Johan & Johan

Redactiesecretariaat:
R. van Zalingen

Strip: Bert Tier
Illustraties: Ben van Mierlo

Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198

Redactie adres:
Postbus 43048
1009 ZA Amsterdam
tel. 020-228871

Listingtelefoon: (ma: 17.00-21.00)
02155-25162

Abonnementen en administratie:
Nicole Balke en Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel. 020-248006
Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Abonnement:
Voor 8 nummers ƒ 47,50 of Bfr. 975 per jaar. Betaling op giro 1585491 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info. Oude nummers kunt U alleen krijgen bij vooruitbetaling van ƒ 6,75 op de bovenstaande rekening. Ook telefonische opgave voor een abonnement is mogelijk. Bel GRATIS 06-02242222 (teleservice), elke dag tot 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze telefoonnummers zijn alleen bedoeld voor opgave van NIEUWE abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Omslagfoto: Infogrames (Operation Neptune)
Zetwerk & druk: NDB, Zoeterwoude
Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud van dit nummer

** EXTRA **



40 pagina's Efficiency Beurs '89 Nieuws.

De belangrijkste computerbeurs van de Lage Landen barst aan het eind van deze maand weer los. In dit extra hart-magazine 40 pagina's informatie en het laatste nieuws over deze grote najaarsbeurs.

Redactioneel

Er is de laatste tijd een trend om spelletjes op de computer als iets voor kinderen te beschouwen. Serieuze software, liefst op een MS-DOS PC, dat is waar iedereen achteraan holt. Vrijwel alle bladen richten zich op die markt, men praat over OS/2 en over steeds snellere PC's met nu al weer de 80486 processor. Zelfs de winkelketens volgen deze trend. Gameworld, toch groot geworden in de C-64 golf, gaat haar naam nu wijzigen in MEGAWORLD. Men wil blijkbaar geen spelletjes-image meer uitstralen.
Dat vind ik eigenlijk wat arrogant, want spelletjes zijn zeker niet minderwaardig. Met name de verschillende Commodore machines, de C-64 voorop, hebben de computer-spelcultuur doen ontstaan. De kwaliteit van een C-64 spel doet zeker niet onder voor een PC- spel en de Amiga slaat de PC met lengtes, zowel in beeld als geluid. Er is wel degelijk behoefte aan een goede spel/video/audio machine, met een serieus accent, en de Commodores zijn daarvan het duidelijkste bewijs.

Luc Sala

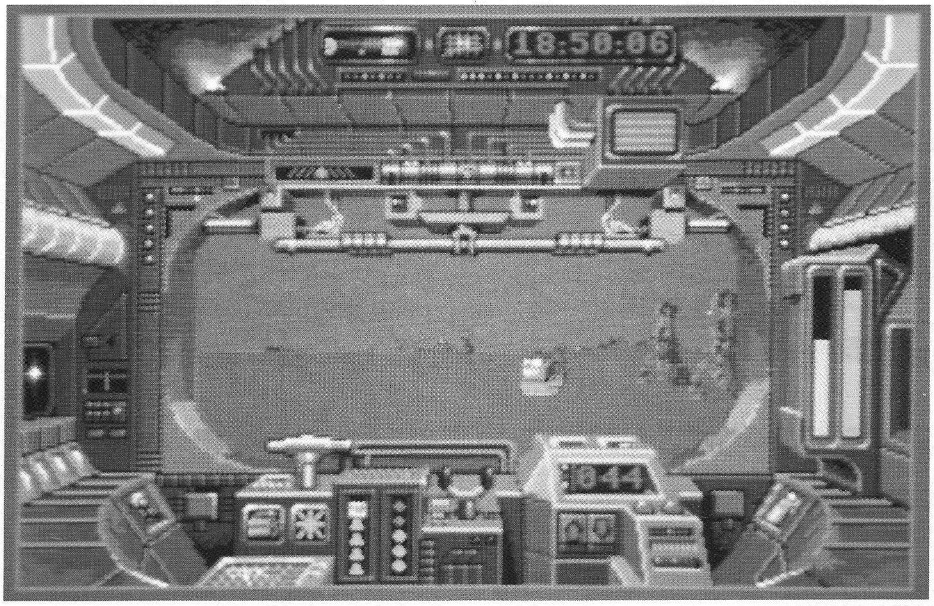
Met enige regelmaat komt er opeens een spel uit, dat van uitstekende kwaliteit is. Operation Neptune is een onderwaterspel van Infogrames, dat de moeite van het bestuderen waard is.

Strijd op de zeebodem

Operation Neptune: spel met fraaie schermbeelden

Infogrames staat al enige tijd bekend als een eerste klas softwarehuis. Niet alleen voor de kwaliteit van het spel, maar ook wegens hun originaliteit. Vanaf volgende week kan ook de Commodore 64 versie van Operation Neptune op deze lijst gezet worden.

In dit spel neem je de rol van Bob Morane op je, in een strijd tegen de gele schaduw. Dit doe je in een 1-mans onderzeeër, uitgerust met de modernste techniek en state-of-the-art bewapening. Om de gele schaduw ervan te weerhouden dat hij zijn macht verspreidt, zul je alle onderwaterstations van hem moeten vernietigen. Dit is op twee manieren mogelijk. De eerste is het beveiligingsschild vernietigen met de kanonnen aan boord van de onderzeer. De tweede manier is er voor te zorgen dat de basis geen energie meer ontvangt. Als de basis geen kracht meer heeft, zal ze zichzelf automatisch vernietigen. Om deze handeling succesvol te volbrengen zul je, gewapend met een harpoen, mijnen en een aqualong, naar zo'n energieaansluiting moeten zwemmen, alwaar de mijn geplaatst kan worden. Hierbij is het wel noodzakelijk dat je de kikvosmannen van de gele schaduw demotiveerd en natuurlijke bewoners zoals haaien en inkvissen ontwijkt.



Op de bodem van de zee...

Geen patroon

De basis van de vijand wordt ook beveiligd door vechters en monteurs, beiden voorzien van gewapende onderwaterjets. Deze kunnen en zullen je te allen tijde proberen te vernietigen, dus zal je ze moeten ontlopen, of ze met je eigen jet moeten afschieten. Nu moet je niet denken dat ze volgens een bepaald patroon varen, niets is minder waar. Als je door de vijand gesignaleerd wordt zal deze zijn koers veranderen en de jacht op je openen. Als je kracht- of zuurstoflijn het nulpunt bereikt, ben je dood, dus dien je er voor te zorgen dat je op tijd bij een thuisbasis bent om supplies in te nemen. Hier kun je ook meteen je voorraad mijnen aanvullen. Als alles vernietigd is kan je gerust boven water komen en een moeilijker level proberen.

fatastische dimensie

Ik merkte dat het *easy level* een goede methode is om alle optie's met een redelijke tegenstand te testen, zodat het geheel interessant blijft. Het spel in combinatie met de muziek maakt het een voortreffelijk spel. De vertraagde beweging die je onder water ervaart geeft het geheel een fantastische dimensie.



Beoordeling:

spel :	**90**
geluid :	**90**
kwaliteit :	**90**
Totaal :	**90**

In deze Tips & Trucs besteden Hylke Sprangers en Michel de Boer uitgebreide aandacht aan de drie logische operatoren en de daarmee verwante boolean expressies. Verder nog een tip over de USR-functie.

Tips & Trucs 64

Lezers worden van harte uitgenodigd om hun eigen Tips & Trucs in te sturen. Maar ook uw vragen of opmerkingen kunt u sturen naar Commodore Info, Postbus 43048, 1009 ZA Amsterdam onder vermelding van Tips & Trucs 64.

Logische operatoren

De Commodore 64 kent de drie logische operatoren AND, OR en NOT. We zullen hieronder uitleggen hoe met deze operatoren gewerkt kan worden. Met deze drie operatoren kunnen we twee kanten op. Dit hangt weer af van de aard van de operanden die meegegeven worden aan de operator. Een operand is een gegeven dat wordt meegegeven aan de operator. De operator voert dan vervolgens een bewerking op of met de operand uit. We maken onderscheid tussen een getal als operand en een expressie als operand. We zullen eerst het geval van een getal als operand behandelen.

Met de AND en OR operator kunnen twee getallen met elkaar vergeleken worden. De uitkomst van deze vergelijking is ook weer een getal. De vergelijking van de twee getallen gebeurt op bit niveau. Elk getal is opgebouwd uit bits, zoals u weet. De twee getallen worden nu bit voor bit met elkaar vergeleken. De uitkomst van een bitvergelijking wordt bepaald door de waarheidstabellen van AND en OR. Deze twee tabellen staan in figuur 1. Bij een vergelijking van twee bits door middel van AND, is de uitkomst alleen een 1, als beide bits een 1 bevatten. Bij OR is de uitkomst een 1, als minstens een van beide bits een 1 bevat. Hier komt een voorbeeldje.

```
getal 1:  170   1 0 1 0 1 0 1 0
getal 2:  204   1 1 0 0 1 1 0 0 AND
          ---   ---------------
uitkomst  136   1 0 0 0 1 0 0 0

getal 1:  170   1 0 1 0 1 0 1 0
getal 2:  204   1 1 0 0 1 1 0 0 OR
          ---   ---------------
uitkomst: 238   1 1 1 0 1 1 1 0
```

In bovenstaand voorbeeld hebben we voor een 8-bit represenentatie gekozen, voor de getallen. In Basic kunt u

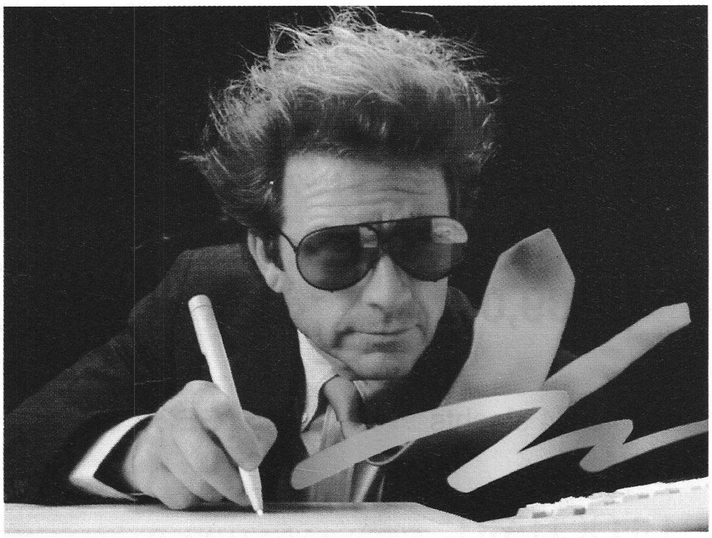


foto:Kurta Corp.

operanden aan AND, OR en NOT meegeven, die tussen -32768 en 32767. De NOT operator heeft in tegenstelling tot AND en OR slechts een operand nodig. De waarheidstabel van NOT staat ook in figuur 1. U kunt zien dat elk bit dat aan staat, met NOT uitgezet wordt, en dat elk bit dat uitstaat aangezet wordt. Ook hiervan een voorbeeldje.

```
getal 1:  170   1 0 1 0 1 0 1 0 NOT
          ---   ---------------
uitkomst:  85   0 1 0 1 0 1 0 1
```

We zeggen dat de NOT-operator een getal inverteert. Bij een acht-bits getal is de uitkomst gelijk aan 255 - getal. U kunt dit ook in het voorbeeld zien: 255 - 170 = 85. De AND en OR operator worden vaak gebruikt om bepaalde bits van geheugenadressen aan of uit te zetten. Met AND kunnen bits makkelijk worden uitgezet, en met OR weer makkelijk aan.

Boolean expressies

We hebben nu gezien hoe we de logische operatoren kunnen gebruiken met getallen als operanden. We gaan nu kijken naar het geval als de operatoren expressies als operanden hebben. Wat een expressie is, kan het beste duidelijk worden gemaakt aan de hand van een voorbeeld.

a = b
8 > 9
a <> 6

De bovenstaande regels bevatten allemaal expressies. We kunnen een expressie het best omschrijven als een relationele vergelijking tussen twee of meer constanten of variabelen. Een expressie kan per definitie waar (TRUE) of onwaar (FALSE) zijn. De tweede expressie bijvoorbeeld (8>9) is FALSE. De eerste expressie (a=b) is alleen TRUE als de variabele a in waarde gelijk is aan variabele b. Bovenstaande expressies worden ook wel Boolean expressies genoemd, naar de briljante Engelse wiskundige George Boole (1815-1864). Met zijn Boolse algebra, waarin de AND, OR en NOT operatoren de basis vormen, wordt hij wel als de grondlegger van de wiskundige logica beschouwd. We weten dus nu dat een expressie altijd of TRUE of FALSE is.

Nu vergelijken we twee expressies met elkaar door middel van AND of OR. Ook nu kunnen we de waarheid-tabellen in figuur 1 gebruiken. Overal waar een 1 staat, denken we aan TRUE en overal waar een 0 staat, denken we aan FALSE. Zo levert een vergelijking tussen twee expressie door middel van AND alleen TRUE op, als de twee expressie zelf ook TRUE zijn. Voor de rest zal de uitkomst FALSE zijn. Bij OR moet tenminste een van de twee expressies TRUE zijn, wil de uitkomst TRUE zijn. Anders is de uitkomst FALSE.
Tenslotte maakt NOT van een ware expressie een onware expressie, en van een onware expressie een ware.
We vinden een directe toepassing van deze theorie bij de IF statements. Het IF statement test een expressie. Als de expressie TRUE oplevert, wordt het gedeelte na de THEN uitgevoerd. Levert de expressie FALSE op, dan wordt het THEN-gedeelte gewoon overgeslagen. Met de AND en OR operator kunnen we nu hele expressies aan elkaar koppelen, en samen in een IF zetten. Een voorbeeld hiervan is:

IF (a>b) AND (a>c) THEN ...

U ziet dat er in een IF statement tegelijk op twee condities wordt getest. In het bovenstaande voorbeeld wordt het THEN gedeelte slechts uitgevoerd, als beide expressies waar zijn. Bekijk nu het volgende voorbeeld eens.

IF (a>b) AND (a>c) OR (b>c) THEN ...

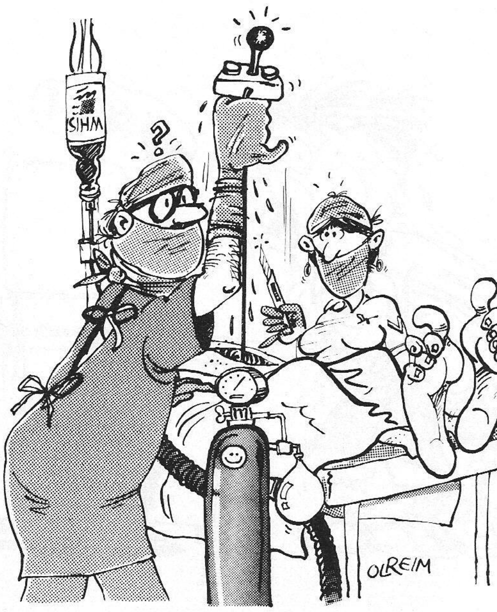
Er zijn nu maar liefst drie expressies aan elkaar gekoppeld met AND en OR. Maar wanneer wordt nu het THEN gedeelte uitgevoerd? Wel, de operator AND heeft een hogere prioriteit dan OR. De computer kijkt dus eerst naar de expressie (a>b) AND (a>c) en bepaalt wat van deze vergelijking de uitkomst is (TRUE of FALSE). Daarna vergelijkt de computer deze verkregen uitkomst door middel van OR met de expressie (b>c). Het THEN gedeelte wordt uitgevoerd als een van de twee expressies (a>b) AND (a>c) of (b>c) TRUE oplevert.
Van de drie operatoren heeft NOT de hoogste prioriteit, daarna AND, en OR heeft de laagste prioriteit. De expressie

(a>b) AND NOT (a>c)

wordt dan ook als volgt behandeld: eerst kijkt de computer naar NOT (a>c). Deze expressie is hetzelfde als (a<=c). Wat we overhouden is dus het volgende: (a>b) AND (a<=c).
Door haakjes te gebruiken, kunnen kan men de prioriteiten veranderen, omdat de gedeelten tussen haakjes als eerste worden berekend.

Getallen als expressies

We hebben het nu gehad over getallen en expressies als operanden. Een expressie is altijd TRUE of FALSE. De computer kan ook een getal op zichzelf, bijvoorbeeld het getal 42, als een expressie beschouwen. Daarbij



wordt nul als FALSE en elk ander getal als TRUE beschouwd. Bekijk het volgende statement eens:

IF 42 THEN ...

Omdat de computer hier 42 als TRUE beschowt, wordt het THEN-gedeelte altijd uitgevoerd. Bij het volgende statement ligt dit echter anders.

IF 42 AND 213 THEN ...

De vergelijking 42 AND 213 levert 0 op. De computer beschouwt 0 als FALSE, en dus wordt het THEN gedeelte niet uitgevoerd. We hebben opgemerkt dat de computer 0 als FALSE en de rest van de getallen als TRUE interpreteert. Dit klopt in het geval als de computer een getal krijgt aangeboden als expressie. Dit getal moet worden geïnterpreteerd als TRUE of als FALSE. In dit geval is 0 FALSE en de rest van de getallen TRUE. Het ligt anders als de computer een expressie heeft (TRUE of FALSE), die als waarde moet worden geïnterpreteerd. Tikt u het volgende maar eens in op de computer:
PRINT (1=2)
De expressie (1=2) levert uiteraard FALSE op. Er staat dus als het ware PRINT FALSE. De computer zal een nul printen, niets vreemds dus. Bij een expressie die TRUE is, zal de computer echter -1 op het beeldscherm printen.
PRINT (1<2)
Op bovenstaand voorbeeld zal de computer antwoorden met -1.

Foutje

In het stukje 'sys met parameters' uit de vorige aflevering hebben we helaas een foutje gemaakt. Het meegeven van numerieke parameters werkt perfect op de manier zoals wij die beschreven hebben. De fout zit echter in het meegeven van string parameters. Het blijkt namelijk dat de computer soms een FORMULA TOO COMPLEX ERROR geeft of een verkeerde string inleest. We hebben vorige keer gezegd dat string parameters met de ROM routine op adres $AD9E konden worden ingelezen. Deze routine zou er dan voor zorgen dat het beginadres van de string in de adressen $22/$23 komt te staan en dat achter de string een 0 wordt gezet, zodat het einde van de string te bepalen is. Dit is ten dele waar. De ROM-routine op $AD9E leest namelijk wel een string parameter in, maar de string wordt daarbij opgeslagen in een speciale stack (stapelgeheugen) voor strings. Nadat een aantal string-parameters met $AD9E is ingelezen, stroomt de stack over en wordt een FORMULA TOO COMPLEX ERROR gegeven. Om dit probleem op te lossen moet direct na de aanroep van de routine op $AD9E, de routine op $B6A6 worden aangeroepen. Deze laatste routine zorgt ervoor dat de stack steeds wordt schoongemaakt. Het beginadres staat dan gewoon in $22/$23. Alleen wordt nu niet achter de string een nul geplaatst, maar staat in de accumulator de lengte van de string. Samengevat kan een string-parameter als volgt worden ingelezen:

```
jsr $ad9e   ;Lees string in
jsr $b6a6
```

Na deze twee instructies bevat de accumulator de lengte van de string en

staat het beginadres van de string in de adressen $22/$23. Bij een routine met 1 string-parameter is het voldoende om de lengte en het startadres van de string te bewaren voor latere verwerking. Als u echter twee of meer string-parameters wilt inlezen, is het niet voldoende om alleen de lengtes en de startadressen van de strings te bewaren. Bij het inlezen van een tweede string-parameter zet de computer de string namelijk over de vorige string parameter heen, waardoor u deze kwijt bent. Om dit te voorkomen, moet u na het inlezen van een string-parameter de hele string naar een ander stuk geheugen verplaatsen (bijvoorbeeld achter het programma).
Numerieke parameters kunnen gewoon op de manier, zoals die in de vorige aflevering beschreven is, worden ingelezen. Hieronder volgt de verbeterde listing om een string op een bepaalde plaats op het scherm af te drukken. Bekijk de vorige aflevering voor een beschrijving van deze routine.

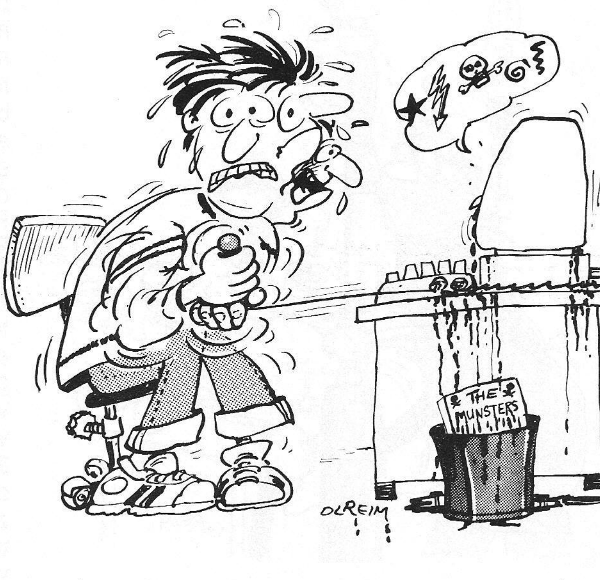
LABELS, MNEMONICS; COMMENTAAR

```
        jsr $aefd  ;skip komma
        jsr $ad9e  ;lees string
        jsr $b6a6
        sta lengte ;bewaar string-
lengte
        lda $22    ;bewaar starta-
dres
        sta $fb
        lda $23
        sta $fc
        jsr $aefd  ;skip komma
        jsr $b79e  ;lees regelnr
        stx regel
        jsr $aefd  ;skip komma
        jsr $ad9e  ;lees kolomnr
        jsr $b7f7
        sty kolom
        ldx regel
        ldy kolom
        clc
        jsr $fff0  ;positioneer cursor
        ldy #$00
lus     lda ($fb),y ;druk string af
        jsr $ffd2  ;druk 1 karakter af
        iny
        cpy lengte
        bne lus
        rts
regel   .byt 0
kolom   .byt 0
lengte  .byt 0
```

Onze excuses voor de gemaakte fout.

De USR instructie

In de vorige aflevering van Tips & Trucs hebben we een methode behandeld waarmee, via de SYS instructie, parameters aan een machinetaal routine kunnen worden meegegeven. Met deze methode was het mogelijk om een routine te schrijven die een meegegeven string op een meegegeven plaats op het scherm zet. Bij deze routine worden alleen waarden vanuit BASIC aan machinetaal doorgegeven. Er bestaan echter ook routines die een bewerking op de meegegeven parameters uitvoeren en de uitkomst vervolgens retourneren aan het BASIC-programma. Er



worden dus niet alleen waarden vanuit BASIC aan machinetaal doorgegeven maar ook vanuit machinetaal aan BASIC. Dergelijke routines worden functies genoemd. Een voorbeeld van een functie, die standaard in de Commodore aanwezig is, is de SQR instructie; deze functie trekt de wortel uit een meegeven parameter en retourneert de uitkomst. De uitkomst kan geretourneerd worden in een variabele zoals in het statement W=SQR(9); variabele W krijgt dan de waarde 3.
Om zelf functies in machinetaal te schrijven, moeten we een methode hebben om waarden vanuit machinetaal aan BASIC door te geven. Met de methode via de SYS instructie is dit niet mogelijk. Maar gelukkig heeft de Commodore een BASIC instructie die dit mogelijk maakt. Dit is de USR instructie. Deze instructie roept, net zoals de SYS instructie, een machinetaal routine aan. Maar in tegenstelling tot de standaard SYS instructie moet er een numerieke parameter worden meegegeven. Deze parameter moet tussen haakjes achter de USR instructie staan. De routine kan dan een bewerking op deze parameter uitvoeren en de uitkomst vervolgens retourneren. De USR instructie is dus een functie en kan bijvoorbeeld als volgt aangeroepen worden: W=USR(10); de machinetaal routine zal dan een bewerking met het getal 10 uitvoeren en de uitkomst in W retourneren.
Uit de aanroep W=USR(10) kunnen we zien dat er enkele verschillen tussen SYS en USR zijn. Een verschil is dat USR aan een variabele kan worden toegekend en SYS niet. Een ander verschil is dat bij SYS het startadres van de machinetaal routine wordt meegegeven en bij USR niet.

startadres

Toch moet de computer ook bij de USR functie weten waar de machinetaal routine begint. We zullen daarom precies bekijken hoe de computer bij de USR instructie te werk gaat. Bij het uitvoeren van een USR functie wordt niet direct naar de zelf gemaakte routine gesprongen (zoals bij de SYS instructie), maar wordt naar adres 784 gesprongen. Op dit adres staat de waarde 76; dit is de stopcode voor de machinetaal JMP-instructie. Het adres waar deze instructie heen springt, staat in de volgende adressen, 785 en 786. De routine die op dit adres staat, wordt dus uiteindelijk uitgevoerd. Om een zelfgeprogrammeerde functie uit te laten voeren, moet het startadres van deze functie in de adressen 785 en 786 worden gepoked. Dit startadres moet in de vorm van een lo- en hi-byte worden gepoked. Het lo-byte moet in 785 worden gepoked en het hi-byte in 786. Voor het uitrekenen van het lo- en het hi-byte van een willekeurig adres kan de volgende formule gebruikt worden:

HI=INT(AD/256):LO=AD-256*HI

Standaard staat in 785 en 786 het adres 45640. Op dit adres begint de routine die de foutmelding '?ILLEGAL QUANTITY ERROR' afdrukt. Als de USR instructie dus wordt gebruikt zonder dat het startadres van de functie in 785 en 786 is gepoked, wordt deze foutmelding gegeven.
Nu we precies weten hoe we een USR functie aan kunnen roepen, gaan we ons bezig houden met het

schrijven van de machinetaal routine. Deze routine zal bestaan uit drie delen:

1. Het inlezen van de meegegeven parameter.
2. Een bewerking met de parameter uitvoeren.
3. Het retourneren van de uitkomst van stap 2.

Nadat de USR functie is aangeroepen, zet de computer de parameter automatisch in een zogenaamde Floating Point Accumulator (FAC). Het werken met Floating Point getallen is een vervelend karwei. We zullen daarom, net zoals in de vorige aflevering, de parameter vertalen naar een 16 bits integer. Voor het vertalen zijn er twee routines beschikbaar. Dit zijn dezelfde routines als de routines uit de vorige aflevering.

De eerste staat op $B1BF. Deze routine vertaalt een parameter die tussen -32768 en 32767 ligt. Het hi-byte wordt in $64 gezet en het lo-byte in $65. De tweede routine staat op $B7F7. Deze routine vertaalt een parameter die tussen 0 en 65535 ligt. Het hi-byte wordt in de accumulator gezet en het lo-byte in het Y-register.

Nadat we de parameter hebben omgezet in een integer, kunnen we deze het beste ergens in het geheugen zetten (bijvoorbeeld achter het eind van het programma).

Nu kunnen we zelf een bewerking met de integer bedenken en programmeren.

Als laatste moeten we de uitkomst van de bewerking retourneren. Dit kan gedaan worden door de uitkomst te vertalen naar een floating point getal en dit getal in FAC te plaatsen. Hierna beëindigen we de routine met RTS. De computer zorgt er dan zelf voor dat de uitkomst aan het BASIC programma wordt geretourneerd. Voor het vertalen van een integer naar floating point is er ook een ROM routine beschikbaar. Deze begint op $B391. Deze routine vertaalt alleen integers tussen -32768 en 32767. Voordat de routine kan worden aangeroepen, moet het hi-byte van de uitkomst in de accumulator worden gezet en het lo-byte in het Y-register. De routine vertaalt de waarde naar een floating point getal en zet deze vervolgens in FAC. Helaas bestaat er geen ROM routine die integers tussen 0 en 65535 naar floating point vertaalt. We hebben echter zelf een kleine subroutine ontwikkeld die dit voor u doet. Als u integers tussen 0 en 65535 wilt retourneren, moet u deze subroutine gewoon opnemen in uw eigen programma. In figuur 2 staat de listing van die routine. Ook bij deze routine moet voor de aanroep de integer in de accumulator en het Y-register worden geplaatst.

Als voorbeeld van een USR functie, geven we nu een eenvoudige assembly listing. De drie eerder genoemde delen waaruit een USR functie altijd bestaat, hebben we in de listing aangegeven met de cijfers 1,2 en 3. Met deze functie kunt u de RAM adressen in de geheugengebieden op $A000-$BFFF en $E000-$FFFF uitlezen. Als parameter moet de USR functie een adres meekrijgen. De waarde van dit



RAM adres wordt geretourneerd. Hierdoor kan dit RAM gebied niet alleen vanuit machinetaal maar ook vanuit BASIC benut worden.

```
MNEMONICS          COMMENTAAR

1 jsr $b7f7        Vertaal parameter
  lda $64
  sta $fc          Bewaar hi-byte van adres
  lda $65
  sta $fb          Bewaar lo-byte van adres
2 sei              Zet interruupt uit
  ldx #$35         Zet ROM uit
  stx $01
  ldy #$00
  lda ($fb),y      Accu wordt adreswaarde
  ldx #$37         Zet ROM aan
  stx $01
  cli              Zet interrupt aan
3 tay              Zet lo-byte in y
  lda #$00         Hi-byte is 0
  jsr $b391        Retourneer waarde
  rts
```

Nadat deze routine in het geheugen is gezet, moet het startadres in 785 en 786 worden gepoked. Stel dat u de routine op adres 49152 zet. Het hi-byte is dan INT(49152/256)=192 en het lo-byte is 49152-256*192=0. U moet dan in 785 een 0 poken en in 786 de waarde 192. Nu kunt u de USR functie aanroepen met bijvoorbeeld PRINT USR (40960). Dit statement zal de waarde van het RAM adres 40960 op het scherm afdrukken. Een aanroep die we al eerder hebben gezien is W=USR(65535). Deze aanroep zorgt er voor dat W de waarde van het RAM adres 65535 krijgt.

Combineren

Met de hierboven beschreven methode is het mogelijk om functies te programmeren die een numerieke parameter hebben en die een waarde retourneren. Door deze methode te combineren met de methode om parameters via het SYS commando mee te geven, kunnen functies gemaakt worden die meerdere numerieke en alfanumerieke parameters hebben en een waarde retourneren. De aanroep van een dergelijke functie zou er dan als volgt uit kunnen zien: W=USR(ASC("E")),"Deze zin bevat vier e's". Deze functie retourneert dan hoe vaak de meegegeven ascii waarde in de meegegeven string voorkomt. In dit voorbeeld wordt W gelijk aan 5. Het eerste wat gedaan moet worden in een dergelijke routine, is het vertalen van de numerieke parameter tussen de haakjes. Dit gaat net zoals bij de USR functie zonder extra parameters. Daarna kunnen de parameters achter de komma worden ingelezen. Dit kan op precies dezelfde manier als het inlezen van de parameters bij het SYS commando (zie daarvoor de vorige aflevering). We gaan hier verder niet op in, maar met de nu verkregen kennis en wat experimenteren, moet u zelf een dergelijke functie kunnen maken. Veel succes ermee!

Michel de Boer & Hylke Spranger

Zo, de vakantie is eindelijk voorbij; de dagen worden weer korter, de regen komt er weer aan en de herfst staat al voor de deur. Kortom, u kunt weer uitgerust en met frisse moed uw Commodore 64 uit de kast halen. Deze vierde aflevering van de cursus Graphics op de 64 zal u in ieder geval voldoende stof geven om weer aan de slag te gaan. De komende maand kunt u met vierkante en tollende ogen achter uw scherm zitten om de fascinerende wereld van high resolution te ontdekken. Get ready to take off. No smoking please!

Graphics op de 64



Deel 4: high resolution

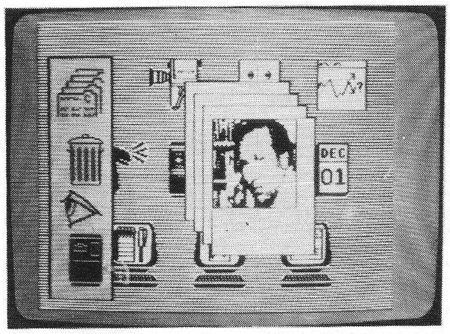
In de vorige afleveringen hebben we de karakters en de sprites uitgebreid behandeld. Het derde en laatste grafische object dat de Commodore 64 rijk is, is nu aangebroken: high resolution. Dit is de grootste van alle grafische objecten. Dit heeft zowel voor- als nadelen. Een van de voordelen is dat er hele beelden mee volgetekend kunnen worden en dat niet, zoals bij karakters, allemaal kleine plaatjes moeten worden samengevoegd tot een groot geheel. Een nadeel is echter dat high resolution minder geschikt is voor animatie, omdat dan veel rekenwerk verricht moet worden.

Deze aflevering zal een beetje aan de theoretische kant blijven, omdat high resolution op de Commodore 64 nu eenmaal een complexe aangelegenheid is. Laat u zich hierdoor geenszins afschrikken! Integendeel, het is een aangename uitdaging om ook dit laatste grafische object onder de knie te krijgen. In ieder geval, veel succes.

De volgende punten zullen in deze aflevering de revue passeren:
* Wat is high resolution
* Opbouw van high resolution
* Hires aanzetten en wissen
* Scherm kleuren
* Een pixel aan zetten
* Applicatie programma

Wat is high resolution?

Het standaard beeld waarmee u normaal werkt wordt low resolution genoemd. De kleinste eenheid van het low resolution scherm is een karakter. Met kleinste eenheid bedoelen we hier het kleinste element op het scherm dat door u geprogrammeerd kan worden. Op de Commodore 64 bestaat nog een andere scherm mode. Deze mode wordt high resolution genoemd. High resolution wordt meestal afgekort tot het acronym hires. In deze aflevering zullen we het dan ook verder hebben over hires. In hires is de kleinste eenheid een pixel



(pixel staat voor picture element). Zoals u wellicht nog weet bestaan karakters en sprites ook uit pixels. In tegenstelling tot sprites en karakters beslaat hires een heel scherm en niet een klein gedeelte van het scherm. Hires is dus een heel scherm dat geheel uit pixels bestaat. Door de verschillende pixels aan of uit te zetten kan men tekeningen over het hele scherm maken. Deze tekeningen kunnen weer in verschillende kleurenmodes worden afgebeeld. Hires is vooral geschikt voor het maken van statische beelden. Omdat een heel scherm vol met pixels een enorme rekencapaciteit vraagt van de computer, is hires minder geschikt voor het maken van dynamische beelden en animaties. Omdat veel rekenwerk relatief veel tijd kost, is snel bewegen van hires er niet bij.

Opbouw van hires

Een hires-scherm bestaat uit 64000 pixels. Horizontaal bestaat het scherm uit 320 pixels en verticaal uit 200 pixels. In totaal zijn dit inderdaad 320 * 200 = 64000 pixels. Dat er 64000 pixels zijn is logisch te beredeneren. Een lores (low resolution) scherm bestaat uit 1000 karakters; 40 karakters in horizontale richting en 25 in verticale richting. Elk van deze karakters is weer opgebouwd uit een raster van 8 bij 8 pixels. In totaal is een karakter dus opgebouwd uit 64 pixels. Het hele scherm bestaat zo uit 1000 * 64 = 64000 pixels.
Het wordt wat afgezaagd, maar ook bij hires wordt een pixel intern weer gerepresenteerd door een bit. Acht horizontale pixels kunnen daardoor door een byte worden weergegeven. Elk pixel van het scherm kan aan (bit aan) of uit (bit uit) staan. Een heel hires-scherm bestaat uit 64000 / 8 = 8000 bytes. Deze bytes moeten natuurlijk weer op een of andere manier geordend zijn in het geheugen. Wel, de byte-ordening van hires is wel logisch, maar niet handig voor de programmeur. We kunnen dit het beste laten zien aan de hand van een plaatje (zie figuur 1). Dit plaatje laat een gedeelte van het hires-scherm zien. Om in te zien hoe de bytes zijn geordend, kunnen we het beste eerst even teruggaan naar lores. Het lores-scherm bestaat uit 1000 karakters.

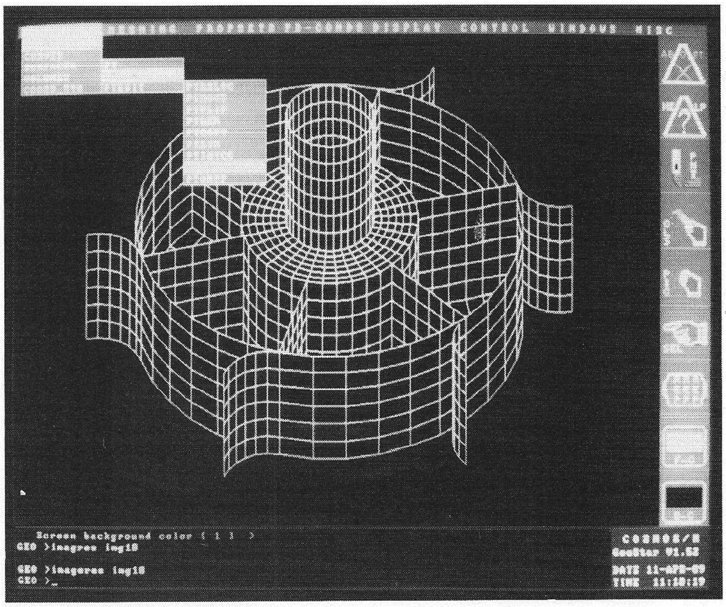
Het karakter linksboven is het eerste karakter, het karakter rechts daarnaast het tweede en het karakter rechtsonder het laatste. Elk karakter correspondeert met een rij van acht verticale bytes op het hires-scherm. Het eerste karakter correspondeert met de eerste acht bytes (byte 0 tot byte 7) van hires; het tweede karakter correspondeert met de tweede acht bytes (byte 8 tot byte 15) enzovoort. Een heel hires-scherm bestaat zo uit 1000 stukjes van elk acht bytes. Elk byte bestaat, zoals gezegd, uit acht bits (lees pixels), die de basiselementen van het hires-scherm vormen. Voor de programmeur is deze byte-ordening niet handig, dit zal nog blijken uit de algoritmen om een willekeurige pixel aan of uit te zetten en lijnen te trekken. Intern was het voor de makers van de Commodore wel makkelijk, omdat hierdoor de opbouw van lores en hires parallel loopt. De opbouw van het hires-scherm zal u op het eerste gezicht wel verwarend doen voorkomen. Het best kunt u figuur 1 goed bestuderen.

Opslaan van hires

Om een hires-plaatje op te slaan is een geheugengebied van 8000 bytes nodig. Deze 8000 bytes kunnen niet op willekeurige plaats van het geheugen opgeslagen worden. In principe komt een aantal verschillende geheugengebieden hiervoor in aanmerking. Om al deze verschillende geheugengebieden te kunnen benutten, moet u eerst iets afweten van de geheugenindeling die de Commodore 64 heeft en het zogenaamde bank switching. Hier zullen we in deze aflevering nog niet op in gaan, maar komen we in een volgende aflevering uitgebreid op terug. In deze aflevering zullen we ons beperken tot het geheugengebied dat begint op adres 8192. In adres 8192 staat de waarde van byte 0, in adres 8193 de waarde van byte 1 en in adres 16191 de waarde van byte 7999. De waarden van bytes worden op dezelfde manier berekend als bij sprites en karakters. Om een adres van een bepaald byte uit te rekenen, moet het bytenummer van de betreffende byte bij 8192 worden opgeteld (zie figuur 1). Zie figuur 2 voor de ordening van de bits in een byte.

Voor- en achtergrondkleur

U heeft maar liefst 64000 pixels tot uw beschikking om een tekening of iets dergelijks mee te maken. Elk van deze pixels kan aan of uit staan. Het hele hires-scherm heeft een achtergrondkleur. Deze kleur kunt u zelf kiezen uit het kleurenpalet van 16 kleuren dat de Commodore u biedt. Als u een pixel van het hires-scherm aanzet, moet deze pixel natuurlijk een kleur hebben. De kleur van de pixels die aanstaan noemen we de voorgrondkleur. Als een pixel uitstaat, heeft dat beeldpunt de gedefinieerde achtergrondkleur.



We zullen nu de term cel introduceren. Een cel op het hires-scherm is een blokje van 8 bytes onder elkaar. In totaal bestaat een hires-scherm uit 1000 cellen. Het verband met lores zult u inmiddels wel gezien hebben. Het lores-scherm bestaat immers ook uit 1000 cellen, karakters genaamd. Wat bij lores-karakters heten, noemen we bij hires-cellen. Een cel bestaat dus uit 8 * 8 = 64 pixels. De cel is een belangrijk begrip bij hires, omdat de voor- en achtergrondkleur direct verband houden met de 1000 cellen waaruit het hires-scherm is opgebouwd. U zou verwachten dat u voor het hele hires-scherm dezelfde voor- en achtergrondkleur moet kiezen. Dit is niet het geval. U kunt per cel een eigen voor- en achtergronkleur kiezen. Elk rastertje van 8 bij 8 pixels heeft zo zijn eigen voor- en achtergrondkleur.

U kunt de voor- en achtergrondkleur kiezen uit de 16 standaardkleuren (kleur 0 tot kleur 15). 16 kleuren kunnen met 4 bits worden gerepresenteerd (%0000 voor kleur 0 tot en met %1111 voor kleur 15). De voor- en achtergrondkleur hebben samen 8 bits nodig en kunnen zo precies in een byte worden gestopt. Om de voor- en achtergrondkleur vast te leggen voor de 1000 cellen, zijn er evenveel bytes nodig, 1000 dus. Met behulp van figuur 3 kunt u de voor- en achtergrondkleur bepalen voor een bepaalde cel. Dit gaat als volgt. U kiest als voorgrondkleur bijvoorbeeld oranje en als achtergrondkleur bruin. U kijkt dan in de tabel. Oranje als voorgrond heeft waarde 128 en bruin als achtergrond heeft waarde 9. Deze waarden telt u bij elkaar op en de verkregen waarde 137 is specifiek voor die bepaalde cel. Vervolgens moeten de 1000 kleurwaarden van de cellen nog aan de computer woren doorgegeven. Als de computer in de hires-mode staat, dient het geheugengebied 1024 tot 2023 als kleurengeheugen. In dit gebied, waar zich normaal het schermgeheugen bevindt, kunt u de 1000 kleurwaarden zetten. Het is natuurlijk niet moeilijk meer om te raden welke cel met welk geheugenadres in het kleurengeheugen correspondeert.

Hires aanzetten

Om de computer in de hires-mode te brengen, moet bit 5 van adres 53265 worden aangezet. Bijvoorbeeld met POKE 53265,PEEK (53265) AND 2^5. Met POKE 53265, PEEK(53265) AND NOT 2^5 staat de computer weer in lores-mode. Tevens moet bit 3 van adres 53272 worden aangezet met POKE 53272,PEEK (53272) OR 2^3. Dit bit heeft te maken met de plaats van het gebruikte geheugengebied. Daar wij voor het geheugengebied vanaf 8192 hebben gekozen, zullen wij niet in deze aflevering verder ingaan op het al dan niet moeten aanzetten van dit bit. Voor het gebruik van het geheugengebied vanaf adres 8192 moet bit 3 van adres 53272 in ieder geval worden aangezet. Maar over dat soort zaken in een volgende aflevering van dit vervolgverhaal meer.

Het hires-scherm wissen

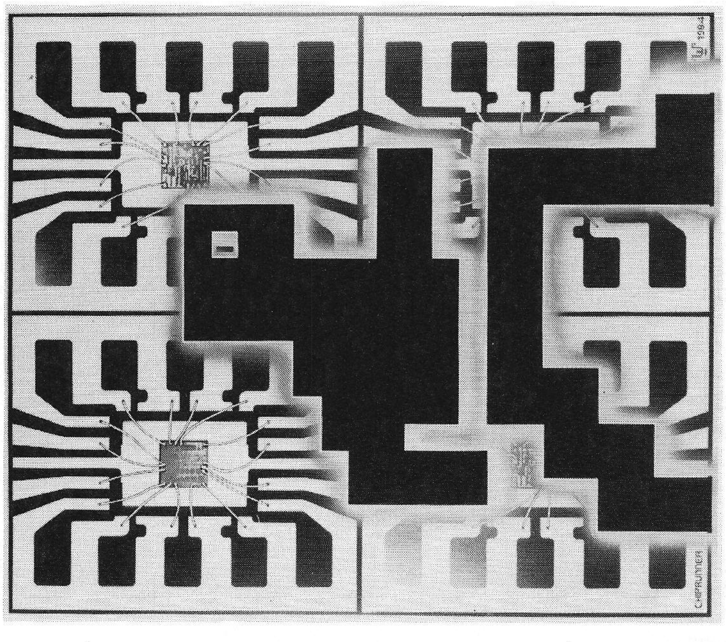
U heeft in de vorige paragraaf gezien hoe u de computer in de hires-mode kan brengen. U heeft hiervoor twee pokes nodig. Als u deze twee pokes gewoon achter elkaar intypt, zult u op uw beeldscherm een heleboel troep waarnemen. Als u dan ook nog wat letters intikt, zullen er nog gekleurde blokjes op het scherm verschijnen. Dit komt omdat het schermgeheugen in het kleurengeheugen van het hires is veranderd. Als u letters intikt, zullen die wel naar geheugengebied 1024 tot 2023 worden geschreven, en zo onstaan de verschillende kleurblokjes op het scherm. Deze kleurblokjes stellen de verschillende cellen van het hires-scherm voor. Omdat de 8000 bytes vanaf adres 8192 in het geheugen ook niet allemaal op nul staan, zullen er ook pixels aan staan, resulterend in de chaos die u op het beeldscherm ziet. Wat we dus moeten doen, is het uitzetten van alle pixels. Met andere woorden, de 8000 bytes die het hires-scherm omvat, moeten op nul worden gezet. Verder is het ook makkelijk om voor het hele scherm een algemene voor- en achtergrondkleur te kiezen. Ook het kleurengeheugen moet dus helemaal volgepoked worden met een bepaalde kleurwaarde. Het volgende korte programma zet het hires-scherm aan en voert de bovengenoemde twee akties uit. Eerst vraagt het programma echter nog om een voor- en achtergrondkleur, die vervolgens worden omgerekend tot een kleurwaarde die in het kleurengeheugen wordt gepoked.

```
10 rem hires-scherm aan en
    wissen
20 input"voorgrondkleur";vg
30 input"achtergrondkleur";ag
40 kw=16*vg+ag
50 poke 53265,peek(53265) or
    2^5
60 poke 53272,peek(53272) or
    2^3
70 for x=1024 to 2023
80 poke x,kw:next
90 for x=8192 to 16191
100 poke x,0:next
120 goto 120
```

U ziet dat de kleurwaarde die in het kleurengeheugen van hires wordt gezet, wordt berekend door: kleurwaarde = 16 * voorgrondkleur + achtergrondkleur. U kunt in de tabel (figuur 3) zien dat de waarden in deze tabel ook echt overeenkomen met deze formule.

Pixels op het scherm

In deze paragraaf zullen we het gaan hebben over het aanzetten van pixels op het hires-scherm. Omdat het hiresscherm voor de programmeur zo onhandig is opgebouwd, is er een hele ingewikkelde formule voor nodig om de juiste pixels aan te zetten. We zullen nu stapje voor stapje uitleggen hoe de formule is opgebouwd. Het volgende stukje is dus tamelijk theoretisch. Het is helemaal niet erg als u het niet helemaal snapt. U kunt zelfs het volgende blok tekst gewoon in zijn geheel overslaan en de formule voor lief nemen, om er daarna gewoon mee te werken. De meer wiskundig aangelegde programmeur zal zich echter in de volgende twee paragrafen verlekkerd bezig kunnen houden met wat hardere theorie.



Het goede byte berekenen

We beschouwen vanaf nu het hiresscherm als een raster van 320 pixels horizontaal en 200 pixels verticaal. Elke pixel op het scherm wordt gedefinieerd door een x- en een y-coördinaat, waarbij de x-coördinaten van 0 tot 319 en de y-coördinaten van 0 tot 199 lopen. Het pixel linksboven heeft zo de coördinaten (0,0) en het pixel rechtsonder (319,199). Als u een pixel wilt aanzetten, moet u eerst de twee coördinaten bepalen. Deze twee coördinaten beschouwen we als de twee variabelen XC (x-coördinaat) en YC (y-coördinaat).

Als er een pixel is uitgekozen, moet het bijbehorende byte en het bijbehorende bit in dat byte worden gezocht. Al eerder in dit artikel hebben we over cellen gesproken, een blok van acht onder elkaar gelegen bytes. We berekenen eerst de cel, waarin het gekozen pixel zich bevindt. Daartoe introduceren we eerst weer twee nieuwe begrippen. Een Celrij is een horizontale rij van cellen en Celkolom is een verticale rij van cellen. Om de cel te berekenen waarin zich het pixel bevindt, berekenen we eerst de celrij (CR) en celkolom (CK) waar de cel zich in bevindt. De twee variabelen CR en CK worden gegeven door de formules:

```
CR = INT(YC/8)
CK = INT(XC/8)
```

Wat we nu gedaan hebben, is het opsplitsen van het hires-scherm in 1000 cellen. Ook deze 1000 cellen beschrijven we in een assenstelsel, van 40 cellen horizontaal en 25 cellen verticaal. U heeft het waarschijnlijk al door, de coördinaten van de cel waarin het pixel zich bevindt dat we willen aanzetten, worden gegeven door (CR, CK).

Van de cel maken we nu weer een stap terug naar de bytes waar de cel uit bestaat. We kunnen het goed begrijpen als u nu al een beetje gek wordt van de verschillende variabelen die we hebben geïntroduceerd. Wel, er komen er nog een paar, te beginnen met FCB. Met de variabel FCB duiden we het eerste byte aan in de gevonden cel. Ook bij FCB hoort er weer een formule, die we maar gelijk geven:

```
FCB = 320*CR + 8*CK
```

De correctheid van deze formule kunt u als volgt inzien. In een Celrij zitten 40 cellen van elk acht bytes. Een celrij beslaat dan 40 * 8 = 320 bytes. Vandaar dat CR met 320 wordt vermenigvuldigd. Dat CK met acht wordt vermenigvuldigd, komt omdat elke cel acht bytes bevat. We hebben nu het eerste byte in de goede cel berekend. Nu moeten we nog het goede byte (BY) in de cel berekenen. Deze stap is redelijk complex en zodoende ook moeilijk om uit te leggen, De formule voor BY luidt in ieder geval:

```
BY = FCB + (YC AND 7)
```

Zoals u ziet wordt in de formule van FCB uitgegaan. Nou komt de moeilij-

ke stap. Het gedeelte (YC AND 7) haalt de laagste drie bits uit de y-coördinaat. Deze drie bits vormen samen een getal tussen de 0 een 7. Dit getal is precies het nummer van het goede byte in de berekende cel, en moet daarom bij FCB worden opgeteld. Overigens is (YC AND 7) hetzelfde als de expressie (YC - 8*INT(YC/8)). Oftewel, (YC AND 7) = (YC - 8*CR). We hopen, dat u deze stap inziet. We zijn er in ieder geval nu bijna met het berekenen van het te gebruiken byte (GBY). Als laatste moet alleen nog het beginadres van het hires-scherm bij BY worden opgeteld. Dit geeft als formule voor GBY:

```
GBY = 8192 + BY
```

We dachten zo dat u na al deze theoretische toestanden wel aan een kopje koffie toe bent. We zijn er echter nog niet helemaal. We hebben weliswaar het goede byte uitgerekend, nu moeten we nog het goede bit in dat byte berekenen, om het tenslotte aan te zetten. Eerst geven we nog even de hele formule voor GBY.

```
GBY = 8192 +
    320*INT(YC/8) +
    8*INT(XC/8) + (YC
    AND 7)
```

Het goede bit berekenen

Nu we eenmaal het goede byte hebben berekend, moeten we nog het goede bit (JBI) berekenen. Daarna kunnen we pas de pixel aan of eventueel uitzetten. Als allerlaatste zullen we nog eenmaal een variabele introduceren: BNO. BNO geeft het bitnummer van het bit aan in het betreffende byte. Hierbij zijn de bits van rechts naar links genummerd, met bit 0 uiterst rechts en bit 7 uiterst links. Ook de volgende formule voor BNO vergt wederom wat inzicht. Hier komt hij:

```
BNO = 7 - (XC AND 7)
```

We zullen het eerst over het laatste deel (XC AND 7) van de formule hebben. Deze expressie extraheert de laagste drie bits uit XC. Ook hier geldt precies hetzelfde voor XC als we destijds over YC hebben gezegd. Het door deze expressie verkregen getal tussen de 0 en de 7 geeft het nummer van het bit weer dat we moeten hebben. Met het verschil dat dit nummer geldt voor een ordening van de bits van links naar rechts. Om deze ordening om te draaien tot de ordening die de computer kent, is de bovenste formule toereikend. Het is dan niet moeilijk om in te zien dat de bovenstaande formule voor BNO werkelijk voldoet. We hebben nu het bitnummer te pakken. Het berekenen van JBI is nu nog een fluitje van een cent. Immers, de waarde van een bit wordt gegeven door de formule: waarde = 2 ^ bitnummer. De formule voor JBI is dan ook als volgt:

```
JBI = 2 ^ BNO
```



We zijn nu eindelijk door al het rekenen en de wiskunde heen. Als u van al deze materie niet zoveel heeft opgestoken, is dat geen enkel probleem. De volgende paragraaf zal weer op het gewone programmeren zijn gericht. Net als de in de vorige paragraaf, besluiten we deze met de totale formule, nu voor JBI

```
JBI = 2^(7 - (XC AND 7))
```

Pixels aanzetten

Na al deze ellendige wiskunde pakken we de draad weer op van het programmeren zelf. We hebben in de vorige twee paragrafen twee formules berekend. Een voor het byte (GBY) waarin het pixel zich bevind. De tweede voor het bit (JBI) zelf in het byte, dat met het uitgekozen pixel correspondeert. Het aanzetten van een pixel kan gebeuren met POKE GBY, PEEK(GBY) OR JBI. Daarentegen kan het uitzetten van een pixel gebeuren met POKE GBY, PEEK(GBY) AND NOT JBI, wat equivalent is met POKE GBY, PEEK(GBY) AND (255 - JBI). Het is nu weer tijd voor een programma. Met het volgende programma kunt u een willekeurig pixel aanzetten. U kunt dus nog niet bijster veel doen met dit programma, maar het laat u wel goed zien hoe het aanzetten van een pixel in de praktijk gebeurt. Het programma maakt eerst nog het scherm schoon. Als voor- en achtergrondkleur worden respectievelijk zwart en wit genomen.

```
10 rem pixel aanzetten
20 input"x-coördinaat
   pixel";xc
30 input"y-coördinaat
   pixel";yc
40 poke
   53265,peek(53265) or
   32
50 poke 53272,
   peek(53272) or 8
60 for x=1024 to 2023
70 poke x,1;next
80 for x=8192 to 16191
90 poke x,0:next
100
   gby=8192+320*int(yc/8)
   +8*int(xc/8)+(yc and
   7)
110 jbi=2^(7-(xc and
    7))
120 poke gby,peek(gby)
    or jbi
130 goto 130
```

Applicatie

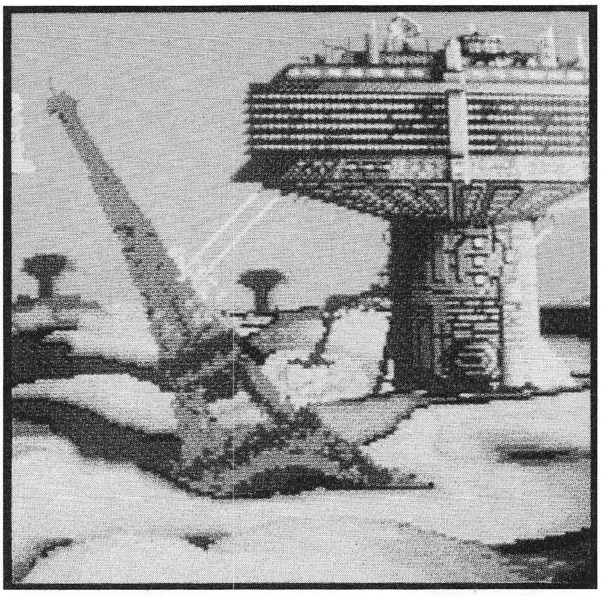
Zoals altijd besluiten we ook deze aflevering met de bespreking van een applicatie. Het programma van deze keer heet "hires routines". Het programma is wat langer dan de programma's uit de vorige afleveringen, maar het is zeker de moeite waard om het over te tikken. Omdat geen normaal mens zulke lange programma's kan overtikken zonder fouten te maken, hebben we achter het programma een checksum afgedrukt. Deze checksum is hetzelfde als de checksum die in de rubriek Print Out wordt gebruikt. Lees die rubriek voor eventuele uitleg.

In het voorgaande hebben we algoritmes besproken om hires te wissen en punten te zetten. Daarbij hebben we de Basic-implementaties van deze algoritmes laten zien. Helaas werkten deze kleine Basic-programma's nogal traag. Om het trekken van lijnen e.d. te versnellen moeten de algoritmes in machinetaal worden geïmplementeerd. Dit is een vrij ingewikkeld karwei; bij het trekken van een lijn zijn de uitkomsten van de berekeningen geen gehele maar gebroken getallen. Het rekenen met zulke getallen in machinetaal is moeilijk en in vergelijking met berekeningen met hele getallen traag. Om toch een hoge snelheid te

halen, zijn er andere ingewikkeldere algoritmes nodig, die we hier niet zullen bespreken. Om u dit werk te besparen, hebben wij een aantal machinetaal routines geschreven.
Het programma "hires routines" bevat zes handige routines. In figuur 4 staan de namen van deze routines met daarachter de startadressen voor gebruik vanuit machinetaal en Basic. In de laatste kolom staat welke parameters er voor elke routine moeten worden meegegeven. Hieronder volgt een korte beschrijving van alle routines:

DOT: Zet een pixel met coördinaten X,Y aan of uit. Of het pixel aan of uit wordt gezet, hangt af van de waarde van de MODE parameter. Als MODE 0 is, wordt er een pixel gewist. Als MODE 1 is, wordt er een pixel aangezet. En als MODE 2 is, wordt een pixel geïnverteerd; d.w.z. als het pixel op coördinaten X,Y aan is, wordt het gewist en als het uit is, wordt het gezet.
LINE: Trek of wis een lijn van punt X1,Y1 naar X2,Y2. Of de lijn getrokken of gewist wordt, hangt weer van de MODE af. Deze MODE heeft dezelfde betekenis als bij DOT.
CLEAR: Wis het hires-scherm.
COLOUR: Kies voor- en achtergrondkleur voor het hele scherm. Als parameters moeten respectievelijk de achter- en de voorgrondkleur worden meegegeven.
TEST: Met deze routine kunt u testen of een bepaald pixel aan of uit staat. Deze routine moet niet, zoals de andere routines, met SYS adres,parameters worden aangeroepen (zie verder), maar met USR(X),Y. Er wordt 0 geretourneerd als het pixel met coördinaten X,Y uit is, als het aan is wordt er 1 geretourneerd. Om bijvoorbeeld te testen of het pixel met coördinaten 300,12 aan staat, kunt u het volgende statement gebruiken: IF USR(300), 12=1 THEN pixel is aan *(zie voor uitleg van de USR instructie de rubriek Tips & Trucs 64.)*
FILL: deze laatste routine kleurt een omsloten vlak in. Als parameters moeten de coördinaten van een willekeurig punt binnen dat vlak en de MODE worden meegegeven. Bij MODE 1 wordt een leeg vlak ingekleurd. Bij MODE 0 wordt een ingekleurd vlak geheel gewist. U moet oppassen dat het vlak dat u wilt inkleuren geheel omsloten is door een onderbroken lijn, anders wordt niet alleen het vlak, maar ook gedeelten buiten het vlak ingekleurd.

Alle routines, behalve TEST, kunnen vanuit Basic met de SYS-instructie worden aangeroepen. De SYS-adressen vindt u in figuur 4 in de kolom 'basic'. De parameters moeten direct achter de SYS-instructie worden meegegeven. Dit gaat als volgt: om bijvoorbeeld een lijn van punt 0,0 naar punt 319,199 in MODE 1 wilt trekken,



tikt u het volgende in: SYS 49186,0, 0,319,199,1 en de computer doet de rest. Het is ook mogelijk om de routines vanuit machinetaal programma's te gebruiken. In de kolom 'machinetaal' vindt u de hexadecimale startadressen van de routines. Om bijvoorbeeld het scherm te wissen gebruikt u JSR $C314. De parameters die moeten worden meegegeven, moeten, voordat u een routine aanroept, in de adressen $02A7 e.v. worden gezet. Een X-coördinaat moet hierbij in een lo- en hi-byte worden gesplitst, omdat deze groter dan 255 kan zijn en daardoor niet meer in een byte past. De TEST routine retourneert in de accumulator een 1 als een pixel aan, een 0 als deze uit staat. Om vanuit machinetaal een punt op coördinaat 10,20 te inverteren, doet u het volgende:

```
lda #10      ;lo-byte
   x-coördinaat
sta $02a7
lda #0       ;hi-byte
   x-coördinaat
sta $02a8
lda #20      ;y-coördinaat
sta $02a9
lda #2       ;MODE
sta $02aa
jsr $c085    ;DOT
```

Alle routines doen hun werk in het hires-scherm dat op adres 8192 start. We hebben geen aparte routines toegevoegd die overschakelen van tekst naar hires en vice versa, omdat dit makkelijk met twee pokes, zoals we eerder hebben uitgelegd, gedaan kan worden.
Voor de machinetaal-programmeurs is het handig om te weten welke geheugenadressen door de hires-routines in beslag worden genomen. Deze zijn de volgende:

```
$A8-$AB, $FB-$FF,
$02A7-$02B6, $0334-
$0339, $C000-$C6FA
en $CE00-$CFFF.
```

Er zijn vele programma's zoals SIMON'S BASIC verkrijgbaar, waarmee hetzelfde als met onze hires-routines gedaan kan worden. Een voordeel van ons programma is dat het weinig geheugen in beslag neemt en dat het niet alleen vanuit Basic maar ook vanuit machinetaal kan worden gebruikt, waardoor er eenvoudig nieuwe routines kunnen worden toegevoegd, zoals een routine om rechthoeken te tekenen. Ook zijn de routines sneller dan in veel andere programma's; onze FILL-routine doet er bijvoorbeeld 22 sec. over om een heel scherm in te kleuren, terwijl SIMON'S BASIC hier 90 sec. over doet. Een klein nadeel is dat u er zelf voor moet zorgen dat de meegegeven coördinaten niet te groot zijn. Als dit wel het geval is, geeft de computer geen foutmelding, maar worden er buiten het hires-geheugen adressen veranderd, waardoor gegevens of programma's verminkt kunnen worden.

Tot zover deze aflevering. In de volgende aflevering gaan we verder met het onderwerp hires. Hierin komen onder andere het trekken van lijnen en cirkels en multi color aan bod.

Michel de Boer & Hylke Sprangers

```
1 rem *****************
2 rem *               *
3 rem * hires routines *
4 rem *               *
5 rem *****************
10 for i= 49152  to 50938
20 read a:poke i,a:next
30 poke 785,115:poke 786,192
1000 data
```

```
32,253,174,32,158,173,32,247
1010 data 183,96,32,0,192,140,167,2
1020 data 141,168,2,32,0,192,140,169
1030 data 2,32,0,192,140,170,2,76
1040 data 133,192,32,0,192,140,167,2
1050 data 141,168,2,32,0,192,140,169
1060 data 2,32,0,192,140,170,2,141
1070 data 171,2,32,0,192,140,172,2
1080 data 32,0,192,140,173,2,76,174
1090 data 192,32,0,192,140,167,2,32
1100 data 0,192,140,168,2,76,54,195
1110 data 32,0,192,140,167,2,141,168
1120 data 2,32,0,192,140,169,2,32
1130 data 0,192,140,170,2,76,134,195
1140 data 76,20,195,32,247,183,140,167
1150 data 2,141,168,2,32,0,192,140
1160 data 169,2,76,88,195,173,170,2
1170 data 10,170,189,238,198,141,54,3
1180 data 189,239,198,141,55,3,173,167
1190 data 2,141,250,198,173,168,2,141
1200 data 251,198,173,169,2,141,252,198
1210 data 32,1,198,108,54,3,173,168
1220 data 2,205,171,2,208,25,173,167
1230 data 2,205,170,2,208,17,173,169
1240 data 2,205,172,2,208,9,173,173
1250 data 2,141,170,2,76,133,192,173
1260 data 173,2,10,170,189,238,198,141
1270 data 54,3,189,239,198,141,55,3
1280 data 169,0,141,0,199,141,1,199
1290 data 173,168,2,205,171,2,144,15
1300 data 208,8,173,170,2,205,167,2
1310 data 176,5,169,1,141,0,199,173
1320 data 172,2,205,169,2,176,5,169
1330 data 1,141,1,199,56,173,0,199
1340 data 208,21,173,170,2,237,167,2
1350 data 141,253,198,173,171,2,237,168
1360 data 2,141,254,198,76,57,193,173
1370 data 167,2,237,170,2,141,253,198
1380 data 173,168,2,237,171,2,141,254
1390 data 198,173,1,199,208,12,173,172
1400 data 2,237,169,2,141,255,198,76
1410 data 83,193,173,169,2,237,172,2
1420 data 141,255,198,173,254,198,240,3
1430 data 76,35,194,173,253,198,205,255
1440 data 198,144,3,76,35,194,173,1
1450 data 199,208,52,173,167,2,141,250
1460 data 198,173,168,2,141,251,198,173
1470 data 169,2,141,252,198,32,1,198
1480 data 173,0,199,208,13,169,63,141
1490 data 52,3,169,198,141,53,3,76
1500 data 208,193,169,82,141,52,3,169
1510 data 198,141,53,3,76,208,193,173
1520 data 170,2,141,250,198,173,171,2
1530 data 141,251,198,173,172,2,141,252
1540 data 198,32,1,198,173,0,199,208
1550 data 13,169,82,141,52,3,169,198
1560 data 141,53,3,76,208,193,169,63
1570 data 141,52,3,169,198,141,53,3
1580 data 173,255,198,74,141,2,199,169
1590 data 0,141,4,199,141,6,199,32
1600 data 166,198,238,4,199,173,255,198
1610 data 205,4,199,144,39,32,99,198
1620 data 173,2,199,24,109,253,198,141
1630 data 2,199,176,5,205,255,198,144
1640 data 222,32,163,198,238,6,199,56
1650 data 173,2,199,237,255,198,141,2
1660 data 199,76,223,193,173,6,199,205
1670 data 253,198,176,6,32,163,198,32
1680 data 166,198,96,173,0,199,208,52
1690 data 173,167,2,141,250,198,173,168
1700 data 2,141,251,198,173,169,2,141
1710 data 252,198,32,1,198,173,1,199
1720 data 208,13,169,99,141,52,3,169
1730 data 198,141,53,3,76,141,194,169
1740 data 120,141,52,3,169,198,141,53
1750 data 3,76,141,194,173,170,2,141
1760 data 250,198,173,171,2,141,251,198
1770 data 173,172,2,141,252,198,32,1
1780 data 198,173,1,199,208,13,169,120
1790 data 141,52,3,169,198,141,53,3
1800 data 76,141,194,169,99,141,52,3
1810 data 169,198,141,53,3,173,253,198
1820 data 141,4,199,173,254,198,141,5
1830 data 199,78,5,199,110,4,199,169
1840 data 0,141,2,199,141,3,199,141
1850 data 7,199,32,166,198,238,2,199
1860 data 208,3,238,3,199,173,253,198
1870 data 205,2,199,176,8,173,3,199
1880 data 205,254,198,240,64,32,63,198
1890 data 173,4,199,24,109,255,198,141
1900 data 4,199,144,3,238,5,199,173
1910 data 5,199,205,254,198,144,203,208
1920 data 8,173,4,199,205,253,198,144
1930 data 193,32,163,198,238,7,199,56
1940 data 173,4,199,237,253,198,141,4
1950 data 199,173,5,199,237,254,198,141
1960 data 5,199,76,170,194,173,7,199
1970 data 205,255,198,176,6,32,163,198
1980 data 32,166,198,96,173,248,198,133
1990 data 251,173,249,198,133,252,24,105
2000 data 32,141,14,199,160,0,152,145
2010 data 251,200,208,251,230,252,166,252
2020 data 236,14,199,208,242,96,169,0
2030 data 133,251,169,4,133,252,173,168
2040 data 2,10,10,10,10,13,167,2
2050 data 160,0,145,251,200,208,251,230
2060 data 252,166,252,224,8,208,243,96
2070 data 173,167,2,141,250,198,173,168
2080 data 2,141,251,198,173,169,2,141
2090 data 252,198,32,1,198,32,210,196
2100 data 176,6,169,0,168,76,124,195
2110 data 169,0,160,1,140,8,199,32
2120 data 145,179,173,8,199,96,173,167
2130 data 2,141,250,198,141,11,199,173
2140 data 168,2,141,251,198,141,12,199
2150 data 173,169,2,141,252,198,141,13
2160 data 199,32,1,198,138,72,173,170
2170 data 2,201,2,208,35,32,210,196
2180 data 176,15,169,210,141,56,3,169
2190 data 196,141,57,3,162,2,76,222
2200 data 195,169,221,141,56,3,169,196
2210 data 141,57,3,162,0,76,222,195
2220 data 10,170,189,244,198,141,56,3
2230 data 189,245,198,141,57,3,189,238
2240 data 198,141,54,3,189,239,198,141
2250 data 55,3,169,0,133,168,133,170
2260 data 169,206,133,169,133,171,104,170
2270 data 169,1,141,9,199,141,10,199
2280 data 32,169,198,144,3,76,170,196
2290 data 206,11,199,173,11,199,201,255
2300 data 208,14,206,12,199,16,9,238
2310 data 12,199,238,11,199,76,51,196
2320 data 32,82,198,32,169,198,144,224
2330 data 32,63,198,238,11,199,208,3
2340 data 238,12,199,32,166,198,173,13
2350 data 199,72,240,33,206,13,199,32
2360 data 120,198,32,169,198,176,14,173
2370 data 9,199,240,14,206,9,199,32
2380 data 166,197,76,90,196,169,1,141
2390 data 9,199,32,99,198,104,141,13
2400 data 199,72,201,199,240,33,238,13
2410 data 199,32,99,198,32,169,198,176
2420 data 14,173,10,199,240,14,206,10
2430 data 199,32,166,197,76,132,196,169
2440 data 1,141,10,199,32,120,198,104
2450 data 141,13,199,238,11,199,208,3
2460 data 238,12,199,173,12,199,240,10
2470 data 173,11,199,201,64,208,3,76
2480 data 170,196,32,63,198,32,169,198
2490 data 144,137,165,168,197,170,208,7
```

```
2500 data 165,169,197,171,208,1,96,32
2510 data 226,197,173,11,199,141,250,198
2520 data 173,12,199,141,251,198,173,13
2530 data 199,141,252,198,32,1,198,76
2540 data 248,195,177,251,61,172,198,240
2550 data 2,56,96,24,96,177,251,61
2560 data 172,198,240,2,24,96,56,96
2570 data 169,0,133,253,169,206,133,254
2580 data 140,8,199,165,251,72,165,252
2590 data 72,138,72,152,72,160,0,177
2600 data 253,141,250,198,200,177,253,141
2610 data 251,198,200,177,253,141,252,198
2620 data 32,1,198,32,169,198,176,29
2630 data 24,165,253,105,3,133,253,144
2640 data 2,230,254,165,253,201,254,208
2650 data 212,165,254,201,207,208,206,172
2660 data 8,199,56,176,23,160,0,173
2670 data 11,199,145,253,200,173,12,199
2680 data 145,253,200,173,13,199,145,253
2690 data 172,8,199,24,104,168,104,170
2700 data 104,133,252,104,133,251,96,165
2710 data 170,201,0,208,8,165,171,201
2720 data 206,208,2,56,96,140,8,199
2730 data 169,0,133,253,169,206,133,254
2740 data 160,0,165,170,197,168,208,27
2750 data 165,171,197,169,208,21,169,0
2760 data 133,170,169,206,133,171,165,253
2770 data 133,168,165,254,133,169,172,8
2780 data 199,24,96,177,170,145,253,230
2790 data 170,208,2,230,171,230,253,208
2800 data 209,230,254,76,114,197,165,168
2810 data 201,254,208,17,165,169,201,207
2820 data 208,11,32,87,197,144,6,32
2830 data 232,196,144,37,96,140,8,199
2840 data 160,0,173,11,199,145,168,200
2850 data 173,12,199,145,168,200,173,13
2860 data 199,145,168,24,165,168,105,3
2870 data 133,168,144,2,230,169,172,8
2880 data 199,96,160,0,177,170,141,11
2890 data 199,200,177,170,141,12,199,200
2900 data 177,170,141,13,199,24,165,170
2910 data 105,3,133,170,144,2,230,171
2920 data 96,173,249,198,133,252,169,0
2930 data 133,251,173,252,198,41,7,168
2940 data 173,252,198,74,74,41,254,170
2950 data 24,189,188,198,101,251,133,251
2960 data 189,189,198,101,252,133,252,173
2970 data 250,198,41,7,170,24,173,250
2980 data 198,41,248,101,251,133,251,173
2990 data 251,198,101,252,133,252,96,232
3000 data 224,8,208,13,162,0,24,165
3010 data 251,105,8,133,251,144,2,230
3020 data 252,96,202,16,13,162,7,56
3030 data 165,251,233,8,133,251,176,241
3040 data 198,252,96,200,192,8,208,15
3050 data 160,0,24,165,251,105,64,133
3060 data 251,165,252,105,1,133,252,96
3070 data 136,16,15,160,7,56,165,251
3080 data 233,64,133,251,165,252,233,1
3090 data 133,252,96,177,251,29,172,198
3100 data 145,251,96,177,251,61,180,198
3110 data 145,251,96,177,251,93,172,198
3120 data 145,251,96,108,52,3,108,54
3130 data 3,108,56,3,128,64,32,16
3140 data 8,4,2,1,127,191,223,239
3150 data 247,251,253,254,0,0,64,1
3160 data 128,2,192,3,0,5,64,6
3170 data 128,7,192,8,0,10,64,11
3180 data 128,12,192,13,0,15,64,16
3190 data 128,17,192,18,0,20,64,21
3200 data 128,22,192,23,0,25,64,26
3210 data 128,27,192,28,0,30,147,198
3220 data 139,198,155,198,221,196,210,196
3230 data 0,32,0
```

```
regel 1     131
regel 2     227
regel 3     215
regel 4     227
regel 5     131
regel 10     46
regel 20     11
regel 30     60
regel 1000    242
regel 1010    88
regel 1020    132
regel 1030    230
regel 1040    128
regel 1050    132
regel 1060    15
regel 1070    20
regel 1080    83
regel 1090    78
regel 1100    94
regel 1110    130
regel 1120    230
regel 1130    134
regel 1140    246
regel 1150    22
regel 1160    106
regel 1170    195
regel 1180    5
regel 1190    186
regel 1200    47
regel 1210    94
regel 1220    133
regel 1230    135
regel 1240    85
regel 1250    132
regel 1260    244
regel 1270    104
regel 1280    90
regel 1290    132
regel 1300    85
regel 1310    95
regel 1320    95
regel 1330    43
regel 1340    133
regel 1350    42
regel 1360    205
regel 1370    191
regel 1380    187
regel 1390    250
regel 1400    152
regel 1410    148
regel 1420    44
regel 1430    52
regel 1440    100
regel 1450    243
regel 1460    48
regel 1470    146
regel 1480    194
regel 1490    49
regel 1500    200
regel 1510    201
regel 1520    182
regel 1530    29
regel 1540    153
regel 1550    151
regel 1560    149
regel 1570    90
regel 1580    6
regel 1590    40
regel 1600    67
regel 1610    163
regel 1620    249
regel 1630    205
regel 1640    207
regel 1650    201
regel 1660    6
regel 1670    205
regel 1680    212
regel 1690    43
regel 1700    188
regel 1710    153
regel 1720    151
regel 1730    203
regel 1740    186
regel 1750    134
regel 1760    38
regel 1770    133
regel 1780    249
regel 1790    90
regel 1800    154
regel 1810    1
regel 1820    199
regel 1830    170
regel 1840    84
regel 1850    167
regel 1860    206
regel 1870    106
regel 1880    251
regel 1890    253
regel 1900    106
regel 1910    43
regel 1920    205
regel 1930    215
regel 1940    203
regel 1950    60
regel 1960    167
regel 1970    1
regel 1980    60
regel 1990    87
regel 2000    179
regel 2010    119
regel 2020    203
regel 2030    36
regel 2040    161
regel 2050    9
regel 2060    248
regel 2070    43
regel 2080    188
regel 2090    194
regel 2100    157
regel 2110    36
regel 2120    16
regel 2130    238
regel 2140    244
regel 2150    239
regel 2160    202
regel 2170    72
regel 2180    194
regel 2190    90
regel 2200    255
regel 2210    87
regel 2220    194
regel 2230    10
regel 2240    3
regel 2250    137
regel 2260    131
regel 2270    148
regel 2280    207
regel 2290    35
regel 2300    145
regel 2310    210
regel 2320    0
regel 2330    150
regel 2340    253
regel 2350    195
regel 2360    49
regel 2370    103
regel 2380    210
regel 2390    155
regel 2400    246
regel 2410    24
regel 2420    175
regel 2430    64
regel 2440    177
regel 2450    195
regel 2460    242
regel 2470    142
regel 2480    6
regel 2490    48
regel 2500    204
regel 2510    97
regel 2520    39
regel 2530    208
regel 2540    100
regel 2550    54
regel 2560    106
regel 2570    34
regel 2580    249
regel 2590    140
regel 2600    129
regel 2610    141
regel 2620    159
regel 2630    230
regel 2640    23
regel 2650    123
regel 2660    98
regel 2670    35
regel 2680    83
regel 2690    243
regel 2700    75
regel 2710    172
regel 2720    101
regel 2730    34
regel 2740    245
regel 2750    250
regel 2760    132
regel 2770    45
regel 2780    47
regel 2790    18
regel 2800    97
regel 2810    79
regel 2820    92
regel 2830    208
regel 2840    233
regel 2850    34
regel 2860    253
regel 2870    191
regel 2880    191
regel 2890    81
regel 2900    37
regel 2910    168
regel 2920    2
regel 2930    247
regel 2940    251
regel 2950    91
regel 2960    147
regel 2970    188
regel 2980    80
regel 2990    82
regel 3000    80
regel 3010    172
regel 3020    89
regel 3030    31
regel 3040    199
regel 3050    175
regel 3060    234
regel 3070    139
regel 3080    230
regel 3090    52
regel 3100    49
regel 3110    55
regel 3120    141
regel 3130    242
regel 3140    240
regel 3150    28
regel 3160    88
regel 3170    186
regel 3180    28
regel 3190    30
regel 3200    32
regel 3210    146
regel 3220    155
regel 3230    160
```

Met een andere kabel en het hierbij gepubliceerde programma krijgt de C-64 gebruiker de beschikking over een supersonische floppy-lader.

KIWI-LOAD (een snellader)

KIWI (the KIngs WIsdom) LOADER is een extreem snelle floppyspeeder: hij kan 202 Blokken in slechts 2.8 (!) seconden binnenladen. Daarmee is KIWI-LOAD het snelste mij bekende laadsysteem! (Turbo Trans buiten beschouwing gelaten omdat die bij iedere diskettewissel de hele flop in de RAM's moet laden, wat veel langer duurt dan die paar tienden van seconden die hij bij het laden zelf sneller is.)

Voor deze snelheid is geen andere formattering nodig: er wordt gewoon het originele GCR-format gebruikt. KIWI-LOAD heeft echter wel een parallelkabel nodig die de (gesokkelde) CIA 6526 met de user-poort verbindt. De kabel (en dus het laadsysteem werkt helaas niet samen met de nieuwe C-128D met de metalen behuizing, omdat de drive erin niet compatible is met de oude 1571 (er zit een customchip in i.p.v. een 6526).

Installatie van het programma

Het programma KIWI-LOAD op de diskette is de RAM-versie van het systeem, bedoeld voor ieder die (zoals ik) geen EPROM-programmer bezit. Het hoeft alleen maar geladen en gestart te worden. Het programma geeft dan een softreset en staat bedrijfsklaar. Voor de ROM-versie moet nu KIWI-ROM geladen en met syss49152 gestart worden. KIWI-LOAD wordt dan aangepast en verschoven naar $2000-$4000. Met een EPROM-programmer kan KIWI-LOAD vervolgens op EPROM gezet worden.

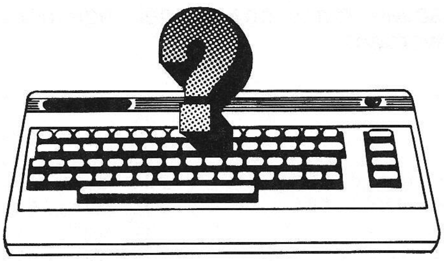
Installatie van de parallelkabel

Deze kabel verbindt de user-poort met de CIA 6526 van de drive. Ik heb voor de CIA 6526 gekozen omdat dat (op de ROM na) het enige gesokkelde IC in de 1570 is. Bovendien is ze er het meest geschikt voor (de 6522's hebben geen NOT PC pin). Voor de kabel heeft men nodig:

- 30cm 10aderig flatcable
- 1 User-poortconnector (+ kapje)
- 1 40 polige IC-voet (gedraaid)

De volgende leidingen moeten met elkaar verbonden worden:

USERPORT	CIA 6526
B (FLAG2)	18 (PC2)
C (PBO)	10 (PBO)
D (PB1)	11 (PB1)
E (PB2)	12 (PB2)
F (PB3)	13 (PB3)
H (PB4)	14 (PB4)
J (PB5)	15 (PB5)
K (PB6)	16 (PB6)
L (PB7)	17 (PB7)
8 (PC2)	24 (FLAG2)



De commando's en functies van KIWI-LOAD

Eerst maar het uit- en aanzetten van:

ROM-Versie uitzetten: poke0,peek (0)or64
RAM-Versie uitzetten: poke1,55 of een hardreset (*het kan dus niet met RUNSTOP/RESTORE of sys64738*)
ROM-Versie weer aanzetten: poke0,peek(0)and 191,RUNSTOP/ RESTORE of een reset.
RAM Versie weer aanzetten: poke1,53

Maar wanneer men een programma heeft geladen dat langer is dan 38K en dus de BASIC is overschreven, dan schakelt de RAM-versie zichzelf uit om een hangup te voorkomen. KIWI-LOAD kan dan herstart worden met sys288 of sys2024. Het in het geheugen aanwezige programma wordt

BLAD 1 Programma kiwi-rom

```
1 rem ***  kiwi-load rom omzetter ***
2 rem *** door bastiaan bakker 1988 ***
3 rem
10 if peek(1)  117 thenprint"laad{S
   PACE}en{SPACE}run{SPACE}eerst{SPAC E}kiwi-load!":end
20 restore
30 fora=0to50:readb:poke49152+a,b:c=c +b:next
 40 if c  8375 thenprint"er{SPACE}zi
   t{SPACE}nog{SPACE}een{SPACE}fout{S
   PACE}in{SPACE}de{SPACE}data{SPACE} !!!":end
50 sys49152
60 print"{SPACE}van{SPACE}$2000{SPACE
   }tot{SPACE}$4000{SPACE}ligt{SPACE} nu{SPACE}de{SPACE}voor"
70 print"{SPACE}eprom{SPACE}geschikte
   {SPACE}verie{SPACE}van{SPACE}kiwi- load"
80 end
100 data160,000,132,251,132,253,169,22 ,4
101 data133,252,169,032,133,254,177,25 1
102 data145,253,200,208,249,230,254,23 ,0
103 data252,208,243,169,055,133,001,16 ,9
104 data231,141,214,061,169,076,141,05 ,3
105 data059,185,055,251,153,055,059,20 ,0
 106 data208,247,096

** EINDE LISTING kiwi-rom
```

hierbij dan wel vernietigd. Voor het laden van disk hoeft men geen ',8' meer in te geven: LOAD 'PRG. NAAM' volstaat. Laden van cassette is met KIWI-LOAD echter niet meer mogelijk. Maar wie bekommert zich daarom als hij ook meer dan 480X sneller zijn programma's van diskette kan laden. Als er geen parallelkabel aanwezig is, schakelt KIWI-LOAD automatisch over op de originele laadroutine. Tijdens het (snel-)laden wordt wel het beeldscherm uitgeschakeld (i.v.m. DMA van de VIC); voor max. 3 seconden stoort dat echter nauwelijks. Met load'$ kan de directory zonder programmaverlies worden opgevraagd. Met de shift-toets kan de directory stilgezet en met runstop afgebroken worden. Meer extra's heeft KIWI-LOAD niet, om de compatibiliteit zo hoog mogelijk te houden.

De snelheid

KIWI-LOAD heeft geen cache-geheugen waarin een hele track gebufferd kan worden. De snelheid is dan ook erg afhankelijk van de volgorde van de te lezen sectors. Daarom even een overzicht van de laadtijden onder verschillende omstandigheden:

(Deze tijden kunnen +/-.1 seconde variëren.)
Met een optimaal geformatteerde schijf: 2.8 seconden, met een met de 1570/71 geformatteerde schijf: 3.1 seconden; met een met de 1541 geformatteerde schijf: 4.6 seconden
De hierbij gebruikte (202 blokken lange) testfile stond aan het begin van de directory en is zonder sector-interleave gesaved (d.w.z. alle sectors direkt achter elkaar). Hiervoor moet voor het saven eerst deze regel ingevoerd worden:

```
open1,8,15:print#1,'u0s'chr$(1)
    :close1
```

Het laden van een file met normale sector-interleave (5) duur ongeveer 10 seconden. In tegenstelling tot vele ander snelladers is er bij KIWI-LOAD niet afgezien van de checksum aan het eind van iedere sector.

Bastiaan Bakker.

Het geheugengebruik

KIWI-LOAD gebruikt de volgende geheugengebieden:

\$100-\$107*	Routine voor het automatisch uitschakelen wanneer de BASIC overschreven is.
\$120-\$128*) \$7E8-\$7F0*)	Restart-Routine (2x)
\$D000-\$D50f*	Initialisatieprogramma van KIWI-LOAD. Wordt via \$120/\$7e8 aangeroepen.
\$F72C-\$FB50	De eigenlijke snellaadroutine.
\$FD6E-\$FD8D	De nieuwe RAM-test.
\$E47E-\$E493	De nieuwe opstartmelding.

* = alleen bij de RAM-versie.

BLAD 1 Programma kiwi

```
  1 rem *** kiwi-load v2 basic-loader ***
  2 rem *** door bastiaan bakker 1988 ***
  3 rem
100 data020,008,196,007,158,050,048,055
101 data048,032,083,085,080,069,082,066
102 data073,084,000,000,000,160,007,185
103 data074,008,153,032,001,153,232,007
104 data136,016,244,120,169,052,133,001
105 data169,082,133,251,169,008,133,252
106 data160,000,132,253,169,208,133,254
107 data177,251,145,253,200,208,249,230
108 data252,230,254,165,254,201,214,208
109 data239,120,169,052,133,001,076,000
110 data208,160,027,185,038,208,153,000
111 data001,136,016,247,032,000,001,032
112 data066,208,160,006,185,031,208,153
113 data000,001,136,016,247,076,000,001
114 data169,053,133,001,076,226,252,169
115 data055,133,001,160,000,169,160,132
116 data251,133,252,177,251,145,251,200
117 data208,249,230,252,208,245,169,052
118 data133,001,096,169,044,160,247,133
119 data253,132,254,169,178,160,208,133
120 data251,132,252,160,000,177,251,145
121 data253,200,208,004,230,252,230,254
122 data192,041,208,241,165,254,201,251
123 data208,235,160,000,185,219,212,153
124 data110,253,200,192,031,208,245,160
125 data000,185,250,212,153,126,228,200
126 data192,021,208,245,169,247,141,249
127 data245,141,183,244,141,251,244,169
128 data054,141,250,244,169,008,141,218
129 data225,169,001,141,220,225,169,229
130 data141,214,253,169,076,141,162,244
131 data169,001,141,163,244,169,251,141
132 data164,244,096,165,186,032,012,237
133 data169,255,076,185,237,036,000,080
134 data003,076,225,255,165,010,208,249
135 data160,000,177,187,201,036,208,007
136 data036,157,016,237,076,140,250,036
137 data000,048,230,104,104,032,044,247
138 data160,000,185,118,248,032,221,237
139 data200,192,039,208,245,032,254,237
140 data160,000,032,044,247,185,157,248
141 data032,221,237,200,192,005,208,245
142 data032,254,237,120,162,010,160,000
143 data136,208,253,202,016,250,169,127
144 data141,013,221,044,013,221,142,003
145 data221,185,162,248,141,001,221,169
146 data016,044,013,221,240,251,200,208
147 data240,185,162,249,141,001,221,169
148 data016,044,013,221,240,251,200,192
149 data240,208,238,169,000,141,003,221
150 data173,017,208,041,239,141,017,208
151 data120,160,000,140,003,221,169,127
152 data141,013,221,044,013,221,032,107
153 data248,072,032,107,248,170,032,107
154 data248,032,107,248,104,208,013,202
155 data202,202,240,005,169,252,076,041
156 data248,076,094,248,162,252,076,252
157 data247,162,254,032,107,248,240,039
158 data032,107,248,160,000,169,016,044
159 data013,221,240,251,173,001,221,145
160 data174,200,202,208,240,032,107,248
161 data208,075,152,024,101,174,133,174
162 data144,215,230,175,076,242,247,032
163 data107,248,170,240,056,202,169,254
164 data133,094,160,000,169,016,044,013
165 data221,240,251,173,001,221,145,174
166 data200,202,208,240,152,072,169,016
167 data196,094,240,013,044,013,221,240
168 data251,200,208,244,032,107,248,208
```

```
169 data012,104,024,101,174,133,174,144
170 data002,230,175,024,036,056,173,017
171 data208,009,016,141,017,208,088,076
172 data170,245,169,016,044,013,221,240
173 data251,173,001,221,096,077,045,087
174 data000,006,046,120,169,127,141,013
175 data064,044,013,064,160,000,140,003
176 data064,169,016,044,013,064,240,251
177 data173,001,064,153,235,004,200,208
178 data240,076,235,004,077,045,069,000
179 data006,169,016,044,013,064,240,251
180 data173,001,064,153,235,005,200,192
181 data240,208,238,076,156,006,169,000
182 data069,022,069,023,069,010,069,011
183 data133,026,160,160,136,208,003,076
184 data079,151,032,084,151,184,080,254
185 data184,173,001,028,201,082,208,236
186 data080,254,184,173,001,028,170,189
187 data013,161,133,082,138,041,001,133
188 data084,080,254,184,173,001,028,133
189 data085,041,240,005,084,170,189,015
190 data159,005,082,197,026,208,197,165
191 data085,041,015,133,085,080,254,184
192 data173,001,028,133,058,041,128,005
193 data085,170,189,029,159,133,082,165
194 data058,170,189,013,162,005,082,197
195 data011,208,161,032,084,151,184,132
196 data031,160,192,080,254,184,173,001
197 data028,201,085,240,003,076,007,151
198 data080,254,184,173,001,028,170,189
199 data013,161,133,082,138,041,001,133
200 data084,080,254,184,173,001,028,133
201 data085,041,240,005,084,170,189,015
202 data159,005,082,133,024,141,001,064
203 data069,031,133,031,165,085,041,015
204 data133,085,080,254,184,173,001,028
205 data133,058,041,128,005,085,170,189
206 data029,159,133,082,165,058,170,189
207 data013,162,005,082,133,025,141,001
208 data064,069,031,133,031,138,041,003
209 data133,058,080,254,184,173,001,028
210 data133,083,041,224,005,058,170,189
211 data042,159,133,082,165,083,170,189
212 data013,163,005,082,141,001,064,069
213 data031,133,031,080,254,184,173,001
214 data028,170,189,013,160,133,082,138
215 data041,007,133,083,080,254,184,173
216 data001,028,133,084,041,192,005,083
217 data170,189,013,159,005,082,141,001
218 data064,069,031,133,031,165,084,170
219 data189,013,161,133,082,138,041,001
220 data133,084,080,254,184,173,001,028
221 data133,085,041,240,005,084,170,189
222 data015,159,005,082,200,240,049,141
223 data001,064,069,031,133,031,165,085
224 data041,015,133,085,080,254,184,173
225 data001,028,133,058,041,128,005,085
226 data170,189,029,159,133,082,165,058
227 data170,189,013,162,005,082,141,001
228 data064,069,031,133,031,076,215,005
229 data197,031,240,003,076,004,151,169
230 data000,141,001,064,165,025,133,011
231 data166,024,240,009,228,010,208,005
232 data134,010,076,000,005,134,010,076
233 data010,151,169,255,141,003,064,169
234 data127,141,013,064,166,024,165,025
235 data134,010,133,011,169,190,133,059
236 data032,078,144,169,230,133,002,036
237 data002,048,252,208,004,165,024,208
238 data242,160,000,140,001,064,169,016
239 data044,013,064,240,251,140,001,064
240 data108,252,255,104,104,169,000,133
241 data144,169,013,032,210,255,032,224
242 data250,032,224,250,032,224,250,170
243 data032,224,250,032,205,189,169,032
244 data032,210,255,032,224,250,240,027
245 data032,210,255,201,034,208,244,032
246 data224,250,240,015,032,210,255,201
247 data034,208,244,169,058,032,210,255
248 data076,184,250,169,013,032,210,255
249 data173,141,002,041,001,208,249,032
250 data225,255,208,186,076,237,250,032
251 data019,238,072,165,144,208,002,104
252 data096,104,104,104,032,066,246,166
253 data045,164,046,096,255,000,234,003
254 data064,001,087,045,077,108,048,003
255 data072,165,000,041,127,133,000,165
256 data186,201,008,144,038,169,000,141
257 data001,221,032,044,247,160,008,185
258 data245,250,032,221,237,136,016,247
259 data032,254,237,173,001,221,201,234
260 data240,009,136,208,246,165,000,009
261 data128,133,000,104,032,254,250,008
262 data072,165,175,201,160,144,018,160
263 data006,185,078,251,153,000,001,136
264 data016,247,076,000,001,169,055,133
265 data001,104,040,096,173,255,159,142
266 data255,159,236,255,159,208,009,232
267 data142,255,159,236,255,159,240,004
268 data160,128,208,002,160,160,141,255
269 data159,162,000,075,073,087,073,045
270 data076,079,065,068,032,086,050,032
271 data032,040,067,041,032,083,080,066
272 rem
500 restore:printchr$(147)
510 fora=0to1375:readb:c=c+b
520 print1375-a"{CRSR-LEFT}{4xSPACE}{CR
    SR-UP}":next
530 ifc  171668 thenprint"er{SPACE}zi
    t{S PACE}nog{SPACE}een{SPACE}fout{
        SPACE}in{SPACE}de{SPACE}data{SPACE
    }!!!":end
540 print"data{SPACE}correct."
600 restore
610 fora=0to1375:readb:poke2049+a,b
620 print1375-a"{CRSR-LEFT}{4xSPACE}{C
    RSR-UP}":next
630 print"data{SPACE}ingelezen."
640 print"save{SPACE}kiwi-load{SPACE}n
    u{SPACE}op{SPACE}een{SPACE}lege{SP
    ACE}disk."
650 poke45,97:poke46,13:end
```

**** EINDE LISTING kiwi**



Basic miniatuurtjes

Een rubriek van Alex van Maarschalkerwaart
Wegens vakantie uitgevoerd door Aryan van Kampen

Gezocht: Miniatuurtjes

voor de Commodore 64,128,16,4 en Amiga

Stuur al je kleine leuke geinige programmatjes op naar de redactie van Commodore Info, Postbus 43048, 1009 ZA Amsterdam.

Hier volgen een viertal inzendingen van Virgil van Maanen, die leuke geluiden uit de CBM-64 haalt.

Woodpecker

De eerste heet woodpecker en doet zoals verwacht een specht na.

```
  1 rem woodpecker
  2 rem door virgil
  3 rem van maanen
  4 rem    uit
  5 rem culemborg !
  6 rem
 10 e=0
 20 ss=54272
 30 v=ss+24:ad=ss+5:sr=ss+6:w=ss+4
 40 poke v,15
 50 poke ad,0:poke sr,240
 60 poke ss+1,65
 70 poke w,103
 80 poke w,0
 90 poke w,17
100 poke w,0
110 poke w,32
120 poke w,0
130 e=e+1:if e=20 then end
140 goto 20
```

Skenner

De tweede is een skenner die de CBM-64 een scanner laat imiteren, wilt u wel eerst al het glaswerk uit de kamer halen!

```
  1 rem skenner !
  2 rem door virgil
  3 rem van maanen
  4 rem    uit
  5 rem culemborg
  7 rem
 10 for a=1 to 10
 20 s=54272:poke s+24,15
 30 poke  s+5,9:poke s+4,129
 40 k=1
 50 poke s+4,65:poke s+3,8
 60 poke s,0:poke s+1,100
 70 for t=1 to 60:nextt
 80 poke s+4,64:for c=0 to 6
 90 poke s+1,c*15+150
100 next c:poke s+4,0
110 k=k+1:if k=2 then 80
120 for t=1 to 250:next t:next a
```

Tokkel

De derde maakt een tokkel geluid, leuk effect voor eigen spelletjes te gebruiken.

```
  1 rem tokkelgeluid
  2 rem door virgil
  3 rem van maanen
  4 rem    uit
  5 rem culemborg
 10 print chr$(147)
 20 ss=54272
 30 for i=ss to ss+24
 40 poke i,0
 50 next i
 60 poke ss+24,15
 70 poke ss+5,0:poke ss+6,249
 80 poke ss,75:poke ss+1,34
 90 poke ss+4,17
100 poke ss+4,129
110 poke ss+4,0
120 for t=1 to 50:next t
130 goto 90
```

Vogelzang

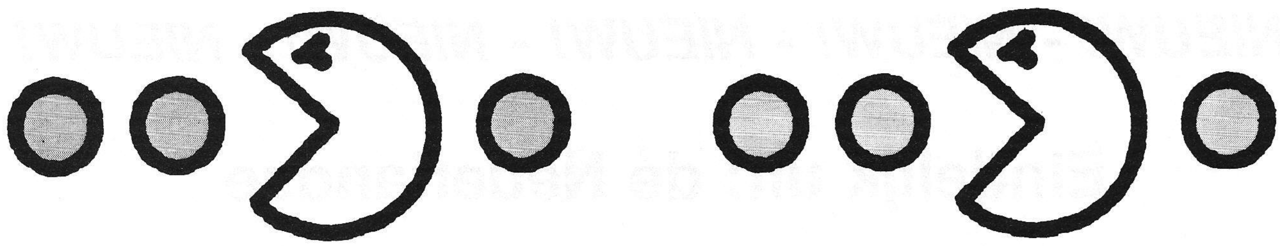
De laatste van Virgil leek na het overnemen van de ingezonden listing meer op een geigerteller,maar na het verbeteren van een foutje in de listing was het wat het zijn moest, vogelenzang.

```
  1 rem vogelenzang
  2 rem door virgil
  3 rem van maanen
  4 rem    uit
  5 rem culemborg
  7 rem
 10 print chr$(147)tab(15)"vogelenzang "
 20 s=54272:poke s+24,15:poke 53280,0:
    poke53281,0:poke 646,1
 30 poke s+5,0:poke s+6,240
 40 poke s+1,65
 50 poke s+4,17
 60 for a=0 to 39:print "-";:next a
 70 ad=rnd(1)*255
 80 if ad<220 then ad=0
 90 poke s+1,ad
100 goto 70
```

Borderflitser

Borderflitser door M.Ramakers laat de border knipperen via de interrupt, dus U kan gewoon verder werken. Door het getal 32 (dit is het 15e in de dataregel) te veranderen in 33 gaat het scherm knipperen. Verander dan wel even de checksum van 2013 naar 2014.

```
  1 rem m.ramakers
 10 for i=0 to 18:read a:poke 49152+i,
    a:x=x+a
 15 next:if x<>2013 then print"datafou
    t":end
 20 sys 49152
 25 data 120,169,13,141,20,3,169,192,1
    41,21,3,88,96,238,32,208,76,49,234
```



Historywriter

Historywriter van Bart Jan Moree laat U zinnen intypen,en na het drukken op shift/return wordt deze zin incl. gebruikte crsr en inst/del toetsen weer op het scherm afgedrukt.

```
 1 rem ***    history writer      ***
 2 rem  ** door bart jan moree   **
 3 rem   *      uit zuidland      *
 4 rem
10 print "{SHIFT CLR}{CTRL 4}":poke 5
   3281,3:poke 53280,3:poke 646,1:i=0
   :y=0
20 poke 204,0:get a$:if a$="" then 20
30 if a$=chr$(141) then 50
40 print a$;:i=i+1:poke 8192+i,asc(a$
   ):goto 20
50 if peek(207) then 50
60 input"{SHIFT CLR}snel{SPACE}of{SPA
   CE}langzaam{SPACE}(s/l)";a$:if a$=
   "l" then y=100
70 print "{SHIFT CLR}":for b=1 to i:p
   rintchr$(peek(8192+b));:for j=0 to
    y:nextj,b
```

vertikaal-scroller

Verticaal-scroller van Bart Jan Moree als variatie op de horizontaal-scrollers.

```
 1 rem *** verticale scrolling ***
 2 rem  ** door bart jan moree **
 3 rem   *     uit  zuidland    *
 4 rem
10 c=1:printchr$(147):poke53281,0:pok
   e53280,0:poke646,1
20 input"max.{SPACE}60{SPACE}karakter
   s{SPACE}";a$(c)
30 get b$:if b$=""then30
40 if b$=chr$(13) then c=c+1:goto60
50 if b$<>chr$(141)then 30
60 printchr$(147):ford=1to24:print:ne
   xt
70 fore=1toc:forf=1tolen(a$(e)):print
   tab(19)mid$(a$(e),f,1):forg=1to100
80 nextg,f,e:end
```

Color-cursor

Color-cursor van Bart Jan Moree verandert tijdens het typen steeds de cursor van kleur wat byv. zeer gekleurde listings geeft.

```
 1 rem ***     color - cursor      ***
 2 rem  ** door bart  jan moree **
 3 rem   *      uit  zuidland      *
 4 rem
10 fori = 49152to49170:reada:pokei,a:
   next:sys49152
20 data 88,169,13,141,20,3,169,192,14
   1,21,3,120,96,238,134,2,76,49,234,
   234
```

Blokfluit

Dit programma van Bart Jan Moree had geen naam meegekregen, maar het leek het meest op een blokfluit, met steeds een ander deuntje.

```
  10 poke 54277,190:poke 54278,25:poke
     54296,15
  20 for j=1 to 8:g=int(8*rnd(1))+1
  30 on g goto 100,101,102,103,104,105,
     106,107
  40 poke 54273,a:poke 54272,b
  50 poke 54276,17:read c:for i=1 to c:
     next
  60 poke 54276,16:next:restore:goto 20
 100 a=53:b=134:goto 40
 101 a=39:b=223:goto 40
 102 a=47:b=107:goto 40
 103 a=53:b=57 :goto 40
 104 a=59:b=190:goto 40
 105 a=71:b=12 :goto 40
 106 a=79:b=191:goto 40
 107 a=71:b=12 :goto 40
1000 data 50,50,200,150,150,50,150,800
```

scroll

Scroll van Benno Goede uit bergen doet het in basic, tekst intypen en 'look your CBM-64 is alive' verander de 100 in regel 21 en 150 als de tekst langer dan 100 karakters moet zijn.

```
  1 rem"{SPACE}gemaakt{SPACE}door{SPAC
    E}benno{SPACE}goede
  2 rem"{SPACE}ruinelaan{SPACE}31{SPAC
    E}bergen{SPACE}(n-h)
  3 rem "--------------
  9 poke 53280,0:poke53281,0:poke646,7
    :printchr$(147)
 10 gosub 110
 20 printchr$(147):poke53272,22
 21 forx=0to100
 30 letu=0
 40 leta=peek(16384+x)
 50 poke1063,a:poke1064,32
 60 letw=peek(1025+u)
 70 poke1024+u,w
 80 u=u+1
 90 if u<41 then60
100 nextx
101 printchr$(147):end
110 input"tekst{SPACE}:";q$
120 forp=1tolen(q$)
130 poke16384+p-1,asc(mid$(q$,p,1))
140 nextp
150 forp=len(q$)to100
160 poke 16384+p,32
170 nextp
180 return
```

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent, dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker da fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW-commando geeft, blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt, SAVE het eerst; mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren, kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4) in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen met de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of is uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
1    rem ******************************
2    rem basic loader "SYNTAX.CHECKSUM"
3    rem na de commando's "run" en "new"
4    rem blijft dit programma in het ge-
5    rem heugen. laad het te testen pro-
6    rem gramma en tik daarna sys 49152.
7    rem ******************************
10   i=49152   :rem beginadres
20   reada:ifa<0then40:rem data ingelezen
30   pokei,a:i=i+1:b=b+a:goto20
40   if b<>16844thenprint"[SHIFT-CLR]fo
     ut[SPACE]in[SPACE]dataregels!":b=0:end
50   poke49184,148:poke49185,192
55   i=49300
60   read a: ifa<0then80
70   pokei,a:b=a+b:i=i+1:goto60
80   if b<>20068thenprint"[SHIFT-CLR]fo
     ut[SPACE]in[SPACE]dataregels![SPAC
     E](vanaf[SPACE]regel[SPACE]240)":b
     =0:end
90   print"data[SPACE]is[SPACE]weggezet"
95   print"checksum[SPACE]testen[SPACE]
     met[SPACE]sys49152"
100  data 165,43,166,44,133,163,134,164
     ,169,147
110  data 32,210,255,160,0,240,3,32,73,192
120  data 32,73,192,208,1,96,32,225,255,208
130  data 3,76,116,164,32,81,192,32,73,192
140  data 240,12,201,32,240,247,24,101,
     167,133
150  data 167,76,37,192,166,167,169,0,1
     32,168
160  data 32,205,189,169,13,32,210,255,
     164,168
170  data 76,17,192,200,208,2,230,164,1
     77,163
180  data 96,162,0,189,123,192,240,6,32,210
190  data 255,232,208,245,32,73,192,170
     ,32,73
200  data 192,132,168,32,205,189,162,3,
     169,32
210  data 32,210,255,202,208,250,169,0,
     133,167
220  data 164,168,96,82,69,71,69,76,32,0
230  data  -1
240  data 165,197,201,3,240,7,201,4,240
250  data 6,76,148,192,76,34,192,169
260  data 147,32,210,255,76,161,192
270  data -1
```

** EINDE LISTING checksum 64 **

Checksum Checksum C-64

REGEL 1	249	REGEL 100	183
REGEL 2	84	REGEL 110	158
REGEL 3	105	REGEL 120	232
REGEL 4	2	REGEL 130	183
REGEL 5	246	REGEL 140	96
REGEL 6	152	REGEL 150	96
REGEL 7	249	REGEL 160	127
REGEL 10	157	REGEL 170	71
REGEL 20	64	REGEL 180	223
REGEL 30	38	REGEL 190	73
REGEL 40	57	REGEL 200	79
REGEL 50	14	REGEL 210	109
REGEL 55	251	REGEL 220	106
REGEL 60	192	REGEL 230	225
REGEL 70	42	REGEL 240	16
REGEL 80	244	REGEL 250	163
REGEL 90	245	REGEL 260	92
REGEL 95	237	REGEL 270	22

Super Lichtkrant

Wel vaker plaatsten wij een lichtkrant, maar de een stelt wat andere eisen dan de ander, daarom heeft de maker van dit programma, Roland van de Klippe, iets extra bedacht. Een tekst loopt midden in het scherm door en loopt ook nog eens dwars door de zijborder heen. Een prachtig staaltje programmeerwerk!

```
100   rem ** super licht krant **
110   rem **    by rolosoft     **
120   rem
130   poke53280,0:poke53281,0:poke646,1
140   print"[SHIFT-CLR][CRSR-DOWN][7xCRS
      R-RIGHT]super-licht-krant"
150   print"[2xCRSR-RIGHT][2xCRSR-DOWN]m
      et[SPACE]deze[SPACE]lichtkrant[SPA
      CE]gaan[SPACE]de[SPACE]letters"
160   print"[2xCRSR-RIGHT]dwars[SPACE]do
      or[SPACE]de[SPACE]zijborder[SPACE]
      heen"
170   print"[2xCRSR-RIGHT]de[SPACE]lette
      rs[SPACE]zelf[SPACE]zijn[SPACE]van
      [SPACE]sprites[8xSPACE]gemaakt"
180   print"[2xCRSR-RIGHT]deze[SPACE]lic
      htkrant[SPACE]start[SPACE]je[SPACE
      ]op[SPACE]met"
190   print"[2xCRSR-RIGHT]sys[SPACE]4915
      2"
200   print"[2xCRSR-RIGHT]stop[SPACE]de[
      SPACE]lichtkrant[SPACE]dmv[SPACE]t
      egelijk"
210   print"[2xCRSR-RIGHT]op[SPACE]de[SP
      ACE]run/stop[SPACE]en[SPACE]restor
      e[SPACE]te[SPACE]drukken"
220   print"[3xCRSR-DOWN][5xCRSR-RIGHT]e
      ven[SPACE]wachten[SPACE]a.u.b."
230   gosub380:print"[3xCRSR-DOWN][8xCRS
      R-RIGHT]<[SPACE]druk[SPACE]op[SPAC
      E]een[SPACE]toets[SPACE]>"
240   geta$:ifa$=""then240
250   print"[SHIFT-CLR][4xCRSR-DOWN][3xC
      RSR-RIGHT]keuze:"
260   print"[10xCRSR-RIGHT]1)lichtkrant[
      SPACE]bekijken"
270   print"[10xCRSR-RIGHT]2)tekst[SPACE
      ]invoeren"
280   geta$:ifa$=""then280
290   ifa$="1"thensys49152
300   ifa$="2"then320
310   goto280
320   input"[SHIFT-CLR]typ[SPACE]text";a
      $
330   fori=1tolen(a$)
340   poke49511+i,asc(mid$(a$,i,1))
350   next
360   poke49511+i,32:poke49512+i,0
370   goto250
380   a=49152
390   readx$:fori=1tolen(x$)/2:x=10*(asc
      (mid$(x$,i*2-1))-65)+asc(mid$(x$,i
      *2))-48
400   pokea,x:a=a+1:next:ifa<49585goto39
      0
410   return
420   data q9u0o1c2u8q9a0o1d2u8o1d3u8d2g
      8w9q9a8n3q7d2g5t2d2c9t2h6c6t2m0q9a
      0o1b4
430   data w0q9m5q2t2o1c0a3o2c1a3q9o9o1b
      8u8q9a1o1c6u8r3b7u8e1m7o1b7u8i8j6q
      2a0s9
440   data i7t3p7a0u8x2w4b7u8y5q9z5o1c1u
      8o1c9u8q9a7q8p3d9u8n6b6z0q2z5q0a7n
      8p3y8
450   data a7u2n6b6y8q0a0p2p3a0g1p3a0g2p
      3a0g3u0u8y4h6h8t3q2a7u2u8z3x4x4q2c
      2d6x4
460   data u6c2u8x8c2u8r2b8u8n6x4p2e1a7a
      9c4o1b7u8x4x4x4x4u2b6w8q9c7o1b7u8q
      9a1o1
470   data c5u8d2s3t2q2y1u2u8z3h6s8z4q2a
      0c4g2a0g1g2u0g3g2t9g3g2t8g3g2n6g3g
      2n5g3
480   data g2n4g3g2h2g3g2h1g3g2h0g3g2a8g
      3g2a7g3g2a6g3g2u0g2g2t9g2g2t8g2g2n
      6g2g2
490   data n5g2g2n4g2g2h2g2g2h1g2g2h0g2g
      2a8g2g2a7g2g2a6g2x2x2x2w4c4u8r3t8q
      7u8h0
500   data q0a0n2z1r7p8b0d8z1b0d8z1b0d8z
      1n3z0q5z1k5v6n3z1q2a0q0a0q9f1n3a1r
      7z0p7
510   data a0g1x2x2x2u0t2a8u8y3n2q7q9f5n
      3a1x0p8u8a2x0p9q0a0r7p8u8a8q9k4q2t
      3n3p8
520   data n4p9j6y1p1e1p1i9p1n7p1s5p1x3p
      1c5p1h3p1t3i3i5i0g9i2d2h6h3g7h2i4d
      2h5i2
530   data g5h8i4d2d2h1g9h7g5g5h5i4d2i6h
      9h9i2d2g7h9h7h7h9g8h9i2g9d2h3h8h0h
      9d2d2
540   data g8h9h9i2d2i2h9h6h9i3h9h0i4e6e
      6e6e6e6d2d2d2d2a0a0a0
```

EINDE LISTING li-krant

Checksum Licht-krant

REGEL 100	186	REGEL 380	248
REGEL 110	74	REGEL 390	84
REGEL 120	143	REGEL 400	78
REGEL 130	243	REGEL 410	142
REGEL 140	41	REGEL 420	130
REGEL 150	254	REGEL 430	150
REGEL 160	32	REGEL 440	30
REGEL 170	56	REGEL 450	160
REGEL 180	213	REGEL 460	232
REGEL 190	27	REGEL 470	128
REGEL 200	16	REGEL 480	224
REGEL 210	13	REGEL 490	61
REGEL 220	59	REGEL 500	164
REGEL 230	16	REGEL 510	237
REGEL 240	99	REGEL 520	106
REGEL 250	201	REGEL 530	9
REGEL 260	144	REGEL 540	116
REGEL 270	75		
REGEL 280	103		
REGEL 290	97		
REGEL 300	84		
REGEL 310	35		
REGEL 320	62		
REGEL 330	202		
REGEL 340	35		
REGEL 350	130		
REGEL 360	68		
REGEL 370	32		

Programma movie

Roland van de Klippe uit Lemelerveld is ook de maker van dit programma, een interrupt programma waarmee het hele beeldscherm beweegt in golven. met als grote bijzonderheid, dat je gewoon door kan gaan met intypen van tekst. Na een paar minuten zo te hebben gewerkt lijkt het als of je verschillende glazen hebt gedronken.

```
10    rem ***********************
20    rem **   movie   screen   **
30    rem *                      *
40    rem **    by rolosoft     **
50    rem ***********************
100   print"[SHIFT-CLR]":poke53280,0:pok
      e53281,0:poke646,1
110   print"[SHIFT-CLR][CRSR-DOWN][4xCRS
      R-RIGHT]**[2xSPACE]movie[SPACE]scr
      een[2xSPACE]by[SPACE][CTRL-9]rolos
      oft[CTRL-0][SPACE]**"
120   print"[2xCRSR-DOWN][2xCRSR-RIGHT]d
      it[SPACE]programma[SPACE]is[SPACE]
      een[SPACE]interrupt[SPACE]pro-"
130   print"[2xCRSR-RIGHT]gramma,[SPACE]
      het[SPACE]werkt[SPACE]dus[SPACE]te
      gelijk[SPACE]met[6xSPACE]basic"
140   print"[CRSR-DOWN][2xCRSR-RIGHT]sta
      rt[SPACE]dit[SPACE]programma[SPACE
      ]met[SPACE]sys[SPACE]49152"
150   print"[5xCRSR-DOWN][5xCRSR-RIGHT]e
      ven[SPACE]wachten[SPACE]voor[SPACE
      ]een[SPACE]demo"
160   gosub1000:print"[CRSR-DOWN][6xCRSR
      -RIGHT]<[SPACE]druk[SPACE]op[SPACE
      ]een[SPACE]toets[SPACE]>"
170   geta$:ifa$=""then170
180   print"[SHIFT-CLR]":fork=1024to2023
      step2:pokek,160:next
190   sys49152
200   print"[HOME][11xCRSR-DOWN][7xCRSR-
      RIGHT]<[SPACE]druk[SPACE]op[SPACE]
      een[SPACE]toets[SPACE]>"
210   geta$:ifa$=""then210
220   end
1000  a=49125
1010  readx$:fori=1tolen(x$)/2:x=10*(asc
      (mid$(x$,i*2-1))-65)+asc(mid$(x$,i
      *2))-48
1020  pokea,x:a=a+1:next:ifa<49346goto10
      10
1025  return
1030  data l4c4b6m9a0a0a0q9t1q0t1d2e0s
      6q5l2k5i0o4a3d2d5s8h6a0w4m0q9f8o1c
      0a3q9
1040  data t2o1c1a3q9a0o1b4w0q9y1o1c6u8q
      9d2o1b8u8r3b7u8e1m7o1b7u8i8x4x4q9t
      9n3q7
1050  data q9h2n3q5q9b2n3q6q9a5n3q9q9g4n
      3q8j6q5q7x4x4x4q6q8r3b8u8u1z5y0c4r
      3b8u8
1060  data u5b8u8y0z1s9m8t2o1c2u8x4x4x4u
      2u8w9q2g4u8w5t8q8u8a4q9g4n3q8q9u0o
      1c2u8
1070  data q9a1o1c5u8q9y1o1c6u8h6e9x4d2a
      2a5a5b6d2u0u0u0u0u1u1u1u1u2u2u2u2u
      3u3u3
1080  data u3u4u4u4u4u4u4u5u5u5u5u5u5u6u
      6u6u6u7u7u7u7u7u7u7u7u6u6u6u6u5u
      5u5u5
1090  data u4u4u4u3u3u3u2u2u2u1u1u1u0u0u
      0u0z5
```

EINDE LISTING movie

Checksum Movie

REGEL 10	85	REGEL 210	96
REGEL 20	119	REGEL 220	128
REGEL 30	227	REGEL 1000	248
REGEL 40	74	REGEL 1010	84
REGEL 50	85	REGEL 1020	111
REGEL 100	157	REGEL 1025	142
REGEL 110	148	REGEL 1030	35
REGEL 120	31	REGEL 1040	206
REGEL 130	233	REGEL 1050	29
REGEL 140	39	REGEL 1060	87
REGEL 150	62	REGEL 1070	201
REGEL 160	218	REGEL 1080	233
REGEL 170	101	REGEL 1090	128
REGEL 180	208		
REGEL 190	163		
REGEL 200	44		

Shoot a skeet

Kleiduiven spelen een hoofdrol in dit spel deze moeten in dit spel uit de lucht geschoten worden. De maker van dit programma is ook al geen onbekende n.l. Roland van der Klippe.

```
10    rem *****************
20    rem * shoot a skeet *
30    rem *  by rolosoft  *
40    rem *****************
50    rem * werkt niet met powercartridg
      e *
100   goto1420
110   sk=2:tg=1:hn$(1)="shoot[SPACE]a[SP
      ACE]skeet":hn$(2)="[2xSPACE]writte
      n[2xSPACE]":hn$(3)="[5xSPACE]by"
120   hn$(4)="[2xSPACE]rolosoft":hn$(5)=
      "[5xSPACE]in":hn$(6)="[4xSPACE]198
      8"
130   hs(1)=23:hs(2)=20:hs(3)=18:hs(4)=1
      6:hs(5)=14:hs(6)=12
140   poke54277,0:poke54278,0:poke54276,
      0
150   poke54296,06:poke54277,1:poke54278
      ,210:poke54276,17
160   poke53280,0:poke53281,0
170   print"[SHIFT-CLR][5xCRSR-DOWN][9xC
      RSR-RIGHT][CTRL-2]--[CTRL-5]shoot[
      SPACE]a[SPACE]skeet[CTRL-2]--"
180   print"[2xCRSR-DOWN][12xCRSR-RIGHT]
      [CTRL-6]by[SPACE]rolosoft"
190   print"[2xCRSR-DOWN][12xSPACE][COM-
      5]instructions"
200   print"[12xSPACE][12xCOM-T]"
210   print"[CRSR-DOWN][CTRL-2]attempt[S
      PACE]to[SPACE]shoot[SPACE]as[SPACE
      ]many[SPACE]skeets[SPACE]as"
220   print"[CRSR-DOWN]possible[SPACE]wi
      th[SPACE]your[SPACE]joystick[SPACE
      ]in[SPACE]port[SPACE]2."
```

```
230  print"[CRSR-DOWN]you[SPACE]have[SP
     ACE]a[SPACE]maximum[SPACE]of[SPACE
     ]25[SPACE]skeets[SPACE]to[SPACE]tr
     y"
240  print"[CRSR-DOWN]and[SPACE]shoot.[
     2xSPACE]good[SPACE]luck[SPACE]!!!"
250  print"[CRSR-DOWN][COM-3]press[SPAC
     E]fire[SPACE]button[SPACE]!!!"
260  if ((peek(56320)and16)<>16)then280
270  nt=int(rnd(1)*255):poke54273,nt:po
     ke54274,nt:goto260
280  poke54296,0:poke54277,0:poke54278,
     0:poke54276,0
290  print"[SHIFT-CLR][2xCRSR-DOWN][CTR
     L 8][6xSPACE]s[SPACE]h[SPACE]o[SPA
     CE]o[SPACE]t[3xSPACE]a[3xSPACE]s[S
     PACE]k[SPACE]e[SPACE]e[SPACE]t"
300  print"[3xCRSR-DOWN][COM-8]the[SPAC
     E]current[SPACE]skill[SPACE]level[
     SPACE]=[SPACE]"sk
310  print"[2xCRSR-DOWN]the[SPACE]curre
     nt[SPACE]target[SPACE]speed[SPACE]
     =[SPACE]"tg
320  print"[2xCRSR-DOWN][COM-3]would[SP
     ACE]you[SPACE]like[SPACE]to[SPACE]
     change[SPACE]this[SPACE]?[SPACE](y
     /n)"
330  get a$:ifa$=""then330
340  ifa$<>"y"and a$<>"n"then330
350  ifa$="n"then540
360  print"[SHIFT-CLR][CRSR-DOWN][CTRL-
     8]1=impossible...2=hard...3=normal"
370  print"[CRSR-DOWN]4=easy...5=like[S
     PACE]a[SPACE]snowflake"
380  print"[2xCRSR-DOWN][CTRL-7]please[
     SPACE]enter[SPACE]new[SPACE]level[
     SPACE](1-5)..."
390  getsk$
400  printtab(30)"[CRSR-UP][CTRL-6][COM
      +]"
410  printtab(30)"[CRSR-UP][CTRL-2][COM
      +]"
420  ifsk$<>""thenprinttab(30)"[CRSR-UP
     ][CTRL-4]"sk$:goto440
430  goto390
440  sk=val(sk$)
450  ifsk<1orsk>5then380
460  print"[SHIFT-CLR][CRSR-DOWN]1=fast
     ...2=medium...3=slowish"
470  print"[2xCRSR-DOWN][CTRL-7]please[
     SPACE]enter[SPACE]new[SPACE]speed[
     SPACE](1-3)..."
480  gettg$
490  printtab(30)"[CRSR-UP][CTRL-6][COM
      +]"
500  printtab(30)"[CRSR-UP][CTRL-2][COM
      +]"
510  iftg$<>""thenprinttab(30)"[CRSR-UP
     ][CTRL-4]"tg$:goto530
520  goto480
530  tg=val(tg$)
540  iftg<1ortg>3then470
550  iftg=1thengt=160:goto580
560  iftg=2thengt=200:goto580
570  gt=255
580  poke828,sk:poke49351,gt
590  print"[SHIFT-CLR][12xCRSR-DOWN][13
     xSPACE][CTRL-5][SPACE]get[SPACE]re
     ady"
600  fort=1to2000:next
610  print"[SHIFT-CLR]":poke53280,3:pok
     e53281,6:gosub1090
620  v=53248:poke2040,203
630  pokev+28,6:pokev+40,0:pokev+41,0:p
     okev+38,0:pokev+37,0:sys49152:poke
     681,0
640  h=1:s=0
650  sys49968
660  n=int(rnd(1)*4)+1
670  ifn=1thenpoke820,0:goto710
680  y=int(rnd(1)*145)+60
690  poke820,y:s=s+1
700  ifn=3thenpoke821,0:goto730
710  y=int(rnd(1)*145)+60
720  poke821,y:s=s+1
730  pokev+21,7
740  sys49168
750  pokev+21,1
760  ifpeek(822)=1thenh=h+1
770  ifpeek(823)=1thenh=h+1
780  ifs>25thenpoke53269,0:h=h-1:goto800
790  goto650
800  poke53280,0:poke53281,0:poke53272,
     21
810  print"[SHIFT-CLR][12xCRSR-DOWN][10
     xSPACE][CTRL-2]you[SPACE]hit[SPACE
     ]"h"[SPACE]skeets"
820  fort=1to2000:next
830  forg=1to6
840  ifh>hs(g)theni=g:goto870
850  nextg
860  ifi=0then890
870  forj=6togstep-1:hn$(j+1)=hn$(j):hs
     (j+1)=hs(j):nextj
880  hs(i)=h
890  z=6:print"[SHIFT-CLR][2xCRSR-DOWN]
     [13xSPACE][COM-6]hall[SPACE]of[SPA
     CE]fame"
900  forh=1to6:print"[2xCRSR-DOWN][CTRL
      7]"h")":next
910  h=1:print"[HOME][2xCRSR-DOWN]"
920  ifh=ithenprint"[2xCRSR-DOWN]":goto
     940
930  printtab(5)"[2xCRSR-DOWN][COM-3]"h
     n$(h)
940  h=h+1
950  ifh>zthen970
960  goto920
970  ifi=0then1040
980  print"[HOME][23xCRSR-DOWN]"tab(12)
     "[CTRL-4]enter[SPACE]your[SPACE]na
     me"
990  print"[HOME][2xCRSR-DOWN]";
1000 forj=1toi:print"[2xCRSR-DOWN]":nex
     tj
1010 input"[CTRL-8][4xCRSR-RIGHT]";hn$(
     i)
1020 iflen(hn$(i))>15thenhn$(i)=left$(h
     n$(i),15)
1030 i=0:goto890
1040 print"[HOME][2xCRSR-DOWN]":forh=1t
     o6:printtab(20)"[2xCRSR-DOWN][CTRL
      6]"hs(h)
1050 next
1060 print"[CRSR-DOWN][CTRL-5][12xSPACE
     ]push[SPACE]fire[SPACE]button"
```

```
1070 if((peek(56320)and16)<>16)then140
1080 goto1070
1090 rem *** draw screen ***
1100 poke53272,29
1110 print"[40xSPACE]";
1120 print"[40xSPACE]";
1130 print"[40xSPACE]";
1140 print"[40xSPACE]";
1150 print"[40xSPACE]";
1160 print"[40xSPACE]";
1170 print"[11xSPACE][CTRL-2]PCDDQ[24xS
     PACE]";
1180 print"[10xSPACE][CTRL-2]FCDDCCQ[3x
     SPACE][CTRL-1]R[6xSPACE]R[3xSPACE]
     R[8xSPACE]";
1190 print"[11xSPACE][COM-5]PQCCDQ[SPAC
     E][CTRL-1]R[SPACE]R[2xSPACE]R[CTRL
     2]FCCQPQ[9xSPACE]";
1200 print"[15xSPACE][CTRL-1]R[2xSPACE]
     R[SPACE]R[3xSPACE]R[16xSPACE]";
1210 print"[17xSPACE][CTRL-1]R[SPACE]R[
     2xSPACE]R[17xSPACE]";
1220 print"[40xSPACE]";
1230 print"[40xSPACE]";
1240 print"[13xSPACE][CTRL-2]JKL[16xSPA
     CE][CTRL-6]PQCCQ[3xSPACE]";
1250 print"[12xSPACE][CTRL-2]IOOOM[15xS
     PACE][CTRL-6]PCDCC[3xSPACE]";
1260 print"[11xSPACE][COM-1]HOOOOON[2xS
     PACE][CTRL-2]GKL[9xSPACE][CTRL-6]P
     CDCQ[3xSPACE]";
1270 print"[10xSPACE][COM-1]GOOOOOOOLFO
     OOM[8xSPACE][CTRL-6]PCDCC[3xSPACE]";
1280 print"[9xSPACE][COM-1]FOOOOOOOOOOO
     OOON[8xSPACE][CTRL-6]D[COM-1]B[CTR
     L 6]D[3xSPACE]"
1290 print"[8xSPACE][COM-1]EOOOOOOOOOOO
     OOOOOL[8xSPACE][COM-1]A[3xSPACE]"
1300 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOO[COM-1]A[CTRL-8]OOOO
     O";
1310 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOO[COM-1]A[CTRL-8]OOOO
     O";
1320 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOO[COM-1]A[CTRL-8]OOOO
     O";
1330 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOO[COM-1]B[CTRL-8]OOOO
     O";
1340 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOO";
1350 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOO";
1360 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOO";
1370 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOO";
1380 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOOO";
1390 print"[CTRL-8]OOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOOOOOOOOOOO[HOME]"
1400 poke2023,79:poke56295,7
1410 return
1420 print"[SHIFT-CLR][CTRL-8][12xCRSR-
     DOWN][5xCRSR-RIGHT]please[SPACE]wa
     it[SPACE]a[SPACE]moment":poke53280
     ,0:poke53281,0
1430 x=49152:ln=1630
1440 gosub3180
1450 v=53248:c=12288:g=0
1460 poke56334,peek(56334)and254
1470 poke1,peek(1)and251
1480 forn=0to511
1490 pokec+n,peek(v+n):next
1500 poke1,peek(1)or4
1510 poke56334,peek(56334)or1
1520 poke53272,(peek(53272)and240)or12
1530 rem
1540 x=12288+520:ln=2730
1550 gosub3180
1560 rem
1570 m=12992:g=0
1580 x=m:ln=2920:gosub3180
1590 m=m+64:x=m:ln=3010:gosub3180
1600 m=m+64:x=m:ln=3100:gosub3180
1610 poke53272,21:goto110
1620 rem *** m/c ***
1630 data169,7,141,21,208,169,100,141,956
1640 data0,208,169,205,141,1,208,96,1028
1650 data173,52,3,141,3,208,173,53,806
1660 data3,141,5,208,173,57,3,141,731
1670 data2,208,173,58,3,141,4,208,797
1680 data173,0,220,41,8,201,8,240,891
1690 data26,173,169,2,201,1,208,3,783
1700 data76,245,194,173,0,208,24,105,1025
1710 data1,201,255,208,3,32,228,194,1122
1720 data141,0,208,173,0,220,41,4,787
1730 data201,4,240,26,173,169,2,201,1016
1740 data0,208,3,76,27,195,173,0,682
1750 data208,56,233,1,201,0,208,3,910
1760 data32,10,195,141,0,208,173,0,759
1770 data220,41,2,201,2,240,15,173,894
1780 data1,208,24,105,1,201,230,208,978
1790 data2,169,229,141,1,208,173,0,923
1800 data220,41,1,201,1,240,18,173,895
1810 data1,208,56,233,1,201,50,208,958
1820 data2,169,51,141,1,208,76,193,841
1830 data192,173,0,220,41,16,201,16,859
1840 data240,3,32,40,193,169,1,141,819
1850 data39,208,173,54,3,201,1,208,887
1860 data8,173,55,3,201,1,208,1,650
1870 data96,169,0,234,105,1,201,160,966
1880 data240,3,76,195,192,173,59,3,941
1890 data24,105,1,141,59,3,205,60,598
1900 data3,240,3,76,28,192,169,0,711
1910 data141,59,3,173,54,3,201,1,635
1920 data240,26,173,61,3,201,1,208,913
1930 data3,76,177,194,173,57,3,24,707
1940 data105,1,201,255,208,3,32,160,965
1950 data194,141,57,3,173,55,3,201,827
1960 data1,240,26,173,62,3,201,1,707
1970 data208,3,76,211,194,173,58,3,926
1980 data56,233,1,201,0,208,3,32,734
1990 data194,194,141,58,3,76,28,192,886
2000 data169,0,141,5,212,141,6,212,886
2010 data141,4,212,169,15,141,24,212,918
2020 data169,5,141,5,212,169,9,141,851
2030 data6,212,169,129,141,4,212,169,1042
2040 data25,141,1,212,173,54,3,201,810
2050 data1,208,3,76,230,193,169,0,880
2060 data141,167,2,141,168,2,173,0,794
2070 data208,205,2,208,144,2,176,22,967
2080 data173,2,208,56,237,0,208,201,1085
2090 data7,176,8,169,1,141,167,2,671
2100 data76,148,193,76,230,193,173,0,1089
```

```
2110 data208,56,237,2,208,201,7,176,1095
2120 data8,169,1,141,167,2,76,148,712
2130 data193,76,230,193,173,1,208,205,1279
2140 data3,208,144,2,176,22,173,3,731
2150 data208,56,237,1,208,201,7,176,1094
2160 data8,169,1,141,168,2,76,202,767
2170 data193,76,230,193,173,1,208,56,1130
2180 data237,3,208,201,7,176,8,169,1009
2190 data1,141,168,2,76,202,193,76,859
2200 data230,193,173,167,2,201,1,240,1207
2210 data3,76,230,193,173,168,2,201,1046
2220 data1,240,3,76,230,193,169,1,913
2230 data141,54,3,32,115,194,173,55,767
2240 data3,201,1,208,1,96,169,0,679
2250 data141,167,2,141,168,2,173,0,794
2260 data208,205,4,208,144,2,176,20,967
2270 data173,4,208,56,237,0,208,201,1087
2280 data7,176,8,169,1,141,167,2,671
2290 data76,40,194,96,173,0,208,56,843
2300 data237,4,208,201,7,176,8,169,1010
2310 data1,141,167,2,76,40,194,96,717
2320 data173,1,208,205,5,208,144,2,946
2330 data176,20,173,5,208,56,237,1,876
2340 data208,201,7,176,8,169,1,141,911
2350 data168,2,76,90,194,96,173,1,800
2360 data208,56,237,5,208,201,7,176,1098
2370 data8,169,1,141,168,2,76,90,655
2380 data194,96,173,167,2,201,1,240,1074
2390 data1,96,173,168,2,201,1,240,882
2400 data1,96,169,1,141,55,3,32,498
2410 data123,194,96,169,205,141,249,7,1184
2420 data76,128,194,169,205,141,250,7,1170
2430 data169,0,141,19,212,141,20,212,914
2440 data141,18,212,169,9,141,19,212,921
2450 data169,3,141,20,212,169,129,141,984
2460 data18,212,169,5,141,15,212,96,868
2470 data169,1,141,61,3,173,16,208,772
2480 data24,105,2,141,16,208,169,0,665
2490 data96,173,57,3,24,105,1,141,600
2500 data57,3,201,88,240,3,76,4,672
2510 data193,96,169,1,141,62,3,173,838
2520 data16,208,56,233,4,141,16,208,882
2530 data169,255,096,173,058,003,056,233,1043
2540 data1,141,58,3,201,0,240,3,647
2550 data76,28,192,96,169,1,141,169,872
2560 data2,173,16,208,24,105,1,141,670
2570 data16,208,169,0,96,173,0,208,870
2580 data24,105,1,141,0,208,201,64,744
2590 data208,5,169,63,141,0,208,76,870
2600 data75,192,169,0,141,169,2,173,921
2610 data16,208,56,233,1,141,16,208,879
2620 data169,254,96,173,0,208,56,233,1189
2630 data1,141,0,208,201,20,208,5,784
2640 data169,21,141,0,208,76,110,192,917
2650 data169,0,141,57,3,169,88,141,768
2660 data58,3,169,0,141,54,3,141,569
2670 data55,3,141,61,3,141,62,3,469
2680 data173,169,2,201,1,208,5,169,928
2690 data5,76,86,195,169,4,141,16,692
2700 data208,169,204,141,249,7,141,250,1369
2710 data7,96,-1,103
2720 rem *** sc graphs ***
2730 data56,56,248,120,56,56,56,56,704
2740 data56,56,56,56,56,124,124,255,783
2750 data205,199,44,165,239,118,208,97,1275
2760 data85,148,106,175,84,40,84,187,909
2770 data1,7,31,15,127,127,127,255,690
2780 data1,3,7,7,103,127,255,255,758
2790 data1,1,1,3,7,15,127,255,410
2800 data1,3,7,15,31,31,127,255,470
2810 data3,7,31,31,63,63,127,255,580
2820 data0,0,0,7,63,127,127,255,579
2830 data1,1,6,199,255,255,255,255,1227
2840 data192,240,248,248,252,252,254,255,1941
2850 data128,128,192,192,192,240,254,255,1581
2860 data192,224,240,248,254,254,255,255,1922
2870 data255,255,255,255,255,255,255,255,2040
2880 data1,7,31,59,51,27,23,15,214
2890 data192,248,200,190,234,104,158,255,1581
2900 data0,0,102,102,24,24,0,0,-1,252
2910 rem *** sprites ***
2920 data0,60,0,1,195,128,6,0,390
2930 data96,24,0,24,32,0,4,32,212
2940 data0,4,64,0,2,64,24,2,160
2950 data128,24,1,128,24,1,128,126,560
2960 data1,128,24,1,128,24,1,64,371
2970 data24,2,64,0,2,32,0,4,128
2980 data32,0,4,24,0,24,6,0,90
2990 data96,1,195,128,0,60,0,-1,480
3000 rem
3010 data0,0,0,0,0,0,0,0,0
3020 data0,0,0,0,0,0,0,0,0
3030 data0,0,0,0,0,255,0,255
3040 data3,186,192,14,174,176,3,170,918
3050 data192,0,255,0,0,0,0,0,447
3060 data0,0,0,0,0,0,0,0,0
3070 data0,0,0,0,0,0,0,0,0
3080 data0,0,0,0,0,0,-1,0
3090 rem
3100 data128,0,4,64,0,4,32,0,232
3110 data118,24,128,0,24,64,0,0,358
3120 data32,192,0,16,192,56,16,0,504
3130 data8,28,28,0,28,16,0,28,136
3140 data0,12,2,0,12,1,0,16,43
3150 data49,48,0,33,16,0,0,0,146
3160 data0,0,0,7,0,0,128,6,141
3170 data0,224,6,3,224,0,3,-1,460
3180 rem
3190 rem
3200 ct=1:ck=0
3210 reada:ct=ct+1:ifa=-1then3290
3220 ifct=10then3260
3230 ifa>255then3280
3240 ck=ck+a
3250 pokex,a:x=x+1:goto3210
3260 ifck<>athenprint"error[SPACE]in[SPACE]line";ln:end
3270 ln=ln+10:goto3200
3280 print"data[SPACE]range[SPACE]error[SPACE]in[SPACE]line";ln:end
3290 reada:ifa<>ckthenprint"error[SPACE]in[SPACE]line[SPACE]";ln:end
3300 return
```

EINDE LISTING shootske

- PRINT OUT - PRINT OUT - PRINT OUT - PRINT OUT - PRINT

Checksum Shootskeet

REGEL 10	89
REGEL 20	45
REGEL 30	246
REGEL 40	89
REGEL 50	168
REGEL 100	80
REGEL 110	147
REGEL 120	255
REGEL 130	106
REGEL 140	104
REGEL 150	113
REGEL 160	37
REGEL 170	110
REGEL 180	140
REGEL 190	76
REGEL 200	129
REGEL 210	110
REGEL 220	134
REGEL 230	219
REGEL 240	55
REGEL 250	118
REGEL 260	98
REGEL 270	131
REGEL 280	159
REGEL 290	122
REGEL 300	129
REGEL 310	19
REGEL 320	41
REGEL 330	99
REGEL 340	56
REGEL 350	116
REGEL 360	95
REGEL 370	157
REGEL 380	38
REGEL 390	99
REGEL 400	97
REGEL 410	72
REGEL 420	245
REGEL 430	37
REGEL 440	40
REGEL 450	131
REGEL 460	250
REGEL 470	29
REGEL 480	96
REGEL 490	97
REGEL 500	72
REGEL 510	239
REGEL 520	37
REGEL 530	34
REGEL 540	123
REGEL 550	244
REGEL 560	240
REGEL 570	233
REGEL 580	161
REGEL 590	45
REGEL 600	218
REGEL 610	105
REGEL 620	102
REGEL 630	94
REGEL 640	154
REGEL 650	178
REGEL 660	254
REGEL 670	75
REGEL 680	164
REGEL 690	35
REGEL 700	80
REGEL 710	164
REGEL 720	36
REGEL 730	93
REGEL 740	170
REGEL 750	87
REGEL 760	225
REGEL 770	226
REGEL 780	76
REGEL 790	36
REGEL 800	136
REGEL 810	126
REGEL 820	218
REGEL 830	133
REGEL 840	2
REGEL 850	201
REGEL 860	254
REGEL 870	205
REGEL 880	47
REGEL 890	118
REGEL 900	79
REGEL 910	119
REGEL 920	212
REGEL 930	233
REGEL 940	29
REGEL 950	37
REGEL 960	36
REGEL 970	34
REGEL 980	119
REGEL 990	77
REGEL 1000	218
REGEL 1010	106
REGEL 1020	182
REGEL 1030	143
REGEL 1040	139
REGEL 1050	130
REGEL 1060	204
REGEL 1070	93
REGEL 1080	81
REGEL 1090	121
REGEL 1100	49
REGEL 1110	24
REGEL 1120	24
REGEL 1130	24
REGEL 1140	24
REGEL 1150	24
REGEL 1160	24
REGEL 1170	9
REGEL 1180	139
REGEL 1190	53
REGEL 1200	240
REGEL 1210	30
REGEL 1220	24
REGEL 1230	24
REGEL 1240	148
REGEL 1250	27
REGEL 1260	166
REGEL 1270	208
REGEL 1280	75
REGEL 1290	33
REGEL 1300	31
REGEL 1310	31
REGEL 1320	31
REGEL 1330	32
REGEL 1340	14
REGEL 1350	14
REGEL 1360	14
REGEL 1370	14
REGEL 1380	14
REGEL 1390	23
REGEL 1400	57
REGEL 1410	142
REGEL 1420	202
REGEL 1430	95
REGEL 1440	89
REGEL 1450	165
REGEL 1460	42
REGEL 1470	127
REGEL 1480	236
REGEL 1490	27
REGEL 1500	28
REGEL 1510	193
REGEL 1520	133
REGEL 1530	143
REGEL 1540	162
REGEL 1550	89
REGEL 1560	143
REGEL 1570	105
REGEL 1580	61
REGEL 1590	206
REGEL 1600	206
REGEL 1610	126
REGEL 1620	74
REGEL 1630	184
REGEL 1640	127
REGEL 1650	28
REGEL 1660	230
REGEL 1670	245
REGEL 1680	226
REGEL 1690	235
REGEL 1700	178
REGEL 1710	117
REGEL 1720	225
REGEL 1730	112
REGEL 1740	192
REGEL 1750	219
REGEL 1760	24
REGEL 1770	15
REGEL 1780	69
REGEL 1790	28
REGEL 1800	17
REGEL 1810	26
REGEL 1820	35
REGEL 1830	74
REGEL 1840	27
REGEL 1850	37
REGEL 1860	125
REGEL 1870	81
REGEL 1880	46
REGEL 1890	234
REGEL 1900	187
REGEL 1910	179
REGEL 1920	17
REGEL 1930	254
REGEL 1940	70
REGEL 1950	34
REGEL 1960	170
REGEL 1970	43
REGEL 1980	170
REGEL 1990	110
REGEL 2000	26
REGEL 2010	123
REGEL 2020	37
REGEL 2030	176
REGEL 2040	9
REGEL 2050	240
REGEL 2060	31
REGEL 2070	83
REGEL 2080	124
REGEL 2090	197
REGEL 2100	198
REGEL 2110	135
REGEL 2120	246
REGEL 2130	239
REGEL 2140	230
REGEL 2150	133
REGEL 2160	248
REGEL 2170	181
REGEL 2180	86
REGEL 2190	44
REGEL 2200	164
REGEL 2210	127
REGEL 2220	234
REGEL 2230	88
REGEL 2240	138
REGEL 2250	31
REGEL 2260	83
REGEL 2270	128
REGEL 2280	197
REGEL 2290	48
REGEL 2300	79
REGEL 2310	247
REGEL 2320	29
REGEL 2330	42
REGEL 2340	31
REGEL 2350	251
REGEL 2360	141
REGEL 2370	201
REGEL 2380	129
REGEL 2390	235
REGEL 2400	145
REGEL 2410	247
REGEL 2420	237
REGEL 2430	115
REGEL 2440	129
REGEL 2450	186
REGEL 2460	92
REGEL 2470	32
REGEL 2480	25
REGEL 2490	229
REGEL 2500	142
REGEL 2510	47
REGEL 2520	84
REGEL 2530	132
REGEL 2540	119
REGEL 2550	109
REGEL 2560	17
REGEL 2570	39
REGEL 2580	12
REGEL 2590	39
REGEL 2600	90
REGEL 2610	87
REGEL 2620	200
REGEL 2630	219
REGEL 2640	130
REGEL 2650	51
REGEL 2660	191
REGEL 2670	132
REGEL 2680	38
REGEL 2690	4
REGEL 2700	31
REGEL 2710	159
REGEL 2720	230
REGEL 2730	49
REGEL 2740	102
REGEL 2750	50
REGEL 2760	159
REGEL 2770	30
REGEL 2780	191
REGEL 2790	17
REGEL 2800	125
REGEL 2810	186
REGEL 2820	136
REGEL 2830	90
REGEL 2840	137
REGEL 2850	137
REGEL 2860	135
REGEL 2870	137
REGEL 2880	78
REGEL 2890	128
REGEL 2900	184
REGEL 2910	181
REGEL 2920	22
REGEL 2930	17
REGEL 2940	172
REGEL 2950	22
REGEL 2960	125
REGEL 2970	171
REGEL 2980	119
REGEL 2990	125
REGEL 3000	143
REGEL 3010	147
REGEL 3020	147
REGEL 3030	107
REGEL 3040	93
REGEL 3050	218
REGEL 3060	147
REGEL 3070	147
REGEL 3080	193
REGEL 3090	143
REGEL 3100	220
REGEL 3110	126
REGEL 3120	178
REGEL 3130	36
REGEL 3140	106
REGEL 3150	228
REGEL 3160	113
REGEL 3170	7
REGEL 3180	143
REGEL 3190	143
REGEL 3200	36
REGEL 3210	198
REGEL 3220	167
REGEL 3230	141
REGEL 3240	185
REGEL 3250	92
REGEL 3260	26
REGEL 3270	121
REGEL 3280	60
REGEL 3290	28
REGEL 3300	142

Programma tafels

Maarten Hoeben uit Amsterdam schreef een paar jaar geleden een programma voor zijn zusje waarmee zij de tafels kon oefenen. Hij heeft nu het programma wat aangepast, zodat het geheel wat overzichtelijker wordt. Het resultaat ziet U hier. Men kan kiezen met welke tafel men wil oefenen. Hierna komt men in het eigenlijke speelveld. De cijfers haal je op door middel van het marsmannetje, dat wordt bestuurd door de joystick in poort 2. Ga met het mannetje voor zo'n cijfer staan. Je pakt het cijfer door de joystick naar onderen te duwen. Vervolgens dient het cijfer naar de geldzak gebracht te worden. Daar wordt het cijfer afgeleverd door de joystick weer naar onderen te laten wijzen. Als een aantal sommen op deze wijze zijn opgelost kom je op een bonusveld terecht. Behalve het marsmannetje loopt er nog een ander raar wezentje rond; als die je pakt kom je achter slot en grendel. Ook mag je niet in het water vallen. Je kan door te springen aan deze gevaren ontkomen. Al met al weer een leuk doe-spel.

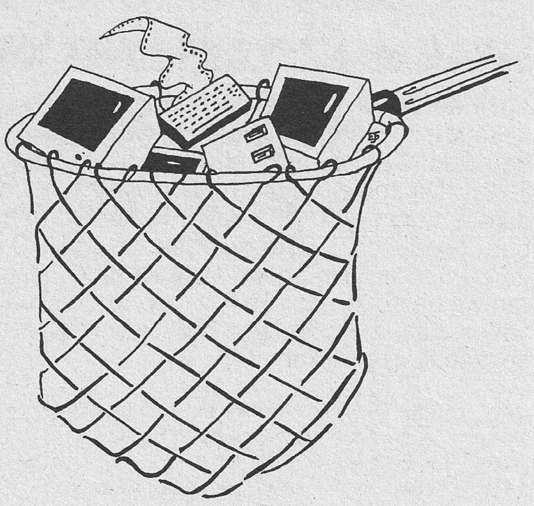
```
100   rem tafels voor de commodore 64
110   rem door : maarten hoeben
120   rem
130   rem         7556 cp  hengelo (o)
140   rem
150   rem voor : commodore info
160   poke657,128:v=53248:poke650,128
170   fort=0to15:pokev+t,0:next
180   rem hoofdprogramma
190   gosub240:rem begin
200   gosub450:rem sprites
210   gosub310:rem cont
220   gosub390:rem spel deel 2
230   end
240   poke53280,11:poke53281,15:print"[S
      HIFT CLR][CTRL-1][CRSR-DOWN][CRSR-
      RIGHT]tafels"
250   print"[2xCRSR-DOWN][CRSR-RIGHT]doo
      r[SPACE]:[SPACE]maarten[SPACE]hoeb
      en"
260   print"[8xCRSR-RIGHT]ir.m.schefferl
      aan[SPACE]37"
270   print"[8xCRSR-RIGHT]7556[SPACE]cp[
      2xSPACE]hengelo(o)"
280   print"[13xCRSR-DOWN][CTRL-9][3xSPA
      CE][2xCOM-'][2xCOM-P][COM-O][COM-P
      ][3xCOM-O][2xCOM-I][CTRL-0][4xCOM-
      U][2xCOM-Y][COM-U][4xCOM-Y][9xCOM-
      T]"
290   print"[CTRL-9][SPACE][CTRL-0][COM-
      V]":print"[CTRL-9][COM-M]":print"[
      CTRL-9][COM-N]":print"[CTRL-9][COM
       N]";
300   return
310   v=53248:pokev+21,1:pokev+28,1:poke
      v+37,1:pokev+38,2:pokev+39,0
320   poke2040,248:pokev+1,190:fort=1to2
      55:pokev,t
330   fork=1to20:next:next:poke2040,245:
      fork=1to500:next:poke2040,246
340   fort=1to200:next:print"[HOME][20xC
      RSR-DOWN]"tab(20)"[CTRL-0][15xSPAC
      E]"
350   fort=190to255:pokev+1,t:next
360   input"[2xCRSR-UP]tafel:";t1
370   ift1<1ort1>255thenprint:goto360
380   return
390   goto540:rem variable
400   gosub960:rem veld 2
410   gosub1140:rem veld 2
420   gosub1260
430   gosub870:p1=peek(v+30):goto650:rem
       spel deel 2
440   end
450   readaa:rem aantal sprites
460   foral=1toaa:readlo:fort=0to62
470   readq:pokelo*64+t,q
480   next:next
490   dimmo(10):fort=1to10:readmo(t):next
500   return
510   print"[HOME][4xCRSR-DOWN]":fort=1t
      o10:print"[5xCRSR-RIGHT][30xSPACE]
      ":next
520   return
530   poke2040,246:fort=1to200:next:poke
      2040,245:return
540   rem programma deel 2 onderdelen
550   rem ***************************
560   poke832,t1:fort=1to10:poke833+t,mo
      (t):next:clr
570   t1=peek(832):fort=1to10:mo(t)=peek
      (833+t):next
580   v=53248:x1=12:y1=24:x2=2.5:dimsp(1
      0):sp(1)=-8:sp(2)=-4:sp(3)=-2:sp(4
      )=-1
590   sp(5)=0:sp(6)=0:sp(7)=1:sp(8)=2:sp
      (9)=4:sp(10)=8:sp=0
600   pl=70:pp=4:r(1)=2:r(2)=4:r(3)=4:r(
      4)=-4:r(5)=-4:r(6)=-2:r(7)=0:r1=11
      2:r2=14
610   r(8)=8:r(9)=-8:r3=48:r4=0:m=1:q=24
      5:pokev+28,3:r=4:m1=0
620   fort=54272to54296:poket,0:next:i4=
      0:pokev+27,0
630   goto400
640   rem besturing
650   x=0:q=245:m=m+1:m1=m1+1
660   if(peek(56320)and2)<>2thengosub1310
670   if(peek(56320)and8)<>8thenx=4:q=248
680   if(peek(56320)and4)<>4thenx=-4:q=2
      47
690   ifx1+x<2orx1+x>230thenx=0
700   if(peek(56320)and16)<>16andp<>32th
      ensp=1:s1=0
710   ifm=11thenm=1
720   ifm1>15thenm1=int(rnd(1)*8)+1:r=r(
      m1):m1=0
730   ifr+r1<=2orr+r1=>232thenr=-r
740   ifr3>166thenlo=2041:gosub900:r1=11
      2:r2=14:r3=48
750   ify1>166thenlo=2040:gosub530:gosub
      900:x1=12:x2=2.5:y1=24
760   x1=x1+x:x2=x2+(x/8):y1=y1+y:pl=pl+
      pp:r1=r1+r:r2=r2+(r/8):r3=r3+r4
770   p=peek(1024+x2+40*(int(y1/8)+2)):p
      2=peek(1024+r2+40*(int(r3/8)+2))
780   y=0:ifp=32theny=4
790   ifsp=1thens1=s1+1:y=sp(s1):ifs1=10
      thensp=0
```

```
800 ifpl>212orpl<70thenpp=-pp
810 pl=peek(v+30):if(pland128)=128 thengosub1310:goto830
820 if(pland1)=1and(pland2)=2and(pland96)<>96thengosub530:goto1380
830 if(pland1)=1and((pland32)=32or(pland64)=64)theny1=pl-50
840 if(pland2)=2and((pland32)=32or(pland64)=64)thenr3=pl-50:m1=16
850 r4=0:ifp2=32thenr4=4
860 gosub870:goto650
870 pokev,22+x1:pokev+1,46+y1:poke2040,q:pokev+2,22+r1:pokev+3,46+r3
880 pokev+11,pl:pokev+13,pl:poke2041,mo(m)
890 return
900 l1=lo-2001:poke54296,15:poke54273,200:poke54272,243:poke54277,16
910 poke54278,232:poke54276,129:pokev+l1,14:pokelo,244:fort=1to100:next
920 pokelo,243:fort=1to500:next
930 pokelo,244:fort=15to1step-.15:poke54296,t:next
940 pokev+l1,0:poke54276,128:iflo=2040thengoto1550
950 return
960 print"[SHIFT-CLR][CTRL-1][3xSPACE][COM-T][COM-Y]ECR[COM-`][2xSPACE]som:"
970 print"[9xSPACE][COM-Y][6xCOM-U][2xCOM-Y][COM-T][COM-Y][4xCOM-U][3xCOM Y][2xCOM-U][COM-Y][4xCOM-T][2xCOM Y][2xCOM-U][COM-Y][COM-T]";
980 print"[14xSPACE]antwoorden[SPACE]=>[2xCRSR-DOWN]"
990 print"[2xCOM-U][2xCOM-Y][COM-T][22xSPACE][CRSR-DOWN][COM-T][3xCOM-Y][COM-U][CTRL-9][COM-O][SPACE][COM-D][CTRL-0][COM-Y][COM-U][COM-Y][COM T]"
1000 print"[CRSR-DOWN][9xSPACE][COM-T][COM-Y][3xCOM-U][2xCOM-Y][3xCOM-U][COM-Y][2xCOM-T][11xSPACE][CRSR-UP][CTRL-9][COM-H][CTRL-0][COM-J][CRSR-DOWN][2xCRSR-LEFT][CTRL-9][COM-J][CTRL-0][COM-H][SPACE]=>[CRSR-DOWN][5xCRSR-LEFT][CTRL-9][COM-G][COM L]";
1010 print"[CRSR-DOWN][5xCRSR-LEFT][CTRL 0][COM-`][COM-O][COM-I][CTRL-9][2xSPACE][CTRL-0][COM-I][COM-O][COM P][COM-`]"
1020 print"[CRSR-DOWN][COM-Y][2xCOM-U][COM-Y][COM-T][23xSPACE][CRSR-UP][COM T][2xCOM-Y][COM-U][COM-Y][2xCOM U][2xCOM-Y][3xCOM-U]";
1030 print"[2xCRSR-DOWN][10xSPACE][COM-T][COM-Y][2xCOM-U][COM-Y][COM-T][2xCOM-Y][COM-U][COM-Y][2xCOM-T][CRSR-DOWN][5xSPACE][COM-T][COM-Y][4xCOM U][COM-Y][COM-T]"
1040 print"[CRSR-DOWN][CTRL-9][SPACE][COM I][CTRL-0][COM-U][COM-Y][COM-T]"
1050 print"[CTRL-9][COM-M][CRSR-DOWN][CRSR-LEFT][COM-N][CTRL-0][8xSPACE][2xCOM-T][2xCOM-Y][4xCOM-U][2xCOM-Y][COM-T][COM-Y][COM-T]"
1060 print"[34xSPACE][COM-D][COM-F][CRSR-DOWN][2xCRSR-LEFT][CTRL-9][COM-H][COM-L][2xCRSR-UP]"
1070 print"[CTRL-9][COM-L][CRSR-DOWN][CRSR-LEFT][CTRL-0][COM-K][CRSR-DOWN][CRSR-LEFT][COM-K][30xSPACE][COM-P][CTRL-9][COM-U][COM-Y][2xSPACE][COM-Y][COM-U][CTRL-0][COM-I][COM-O]";
1080 print"[CTRL-9][COM-7][SPACE][COM-T][2xCOM-Y][3xCOM-U][2xCOM-Y][COM-T][COM-Y][6xCOM-U][COM-Y][COM-U][COM Y][COM-U][COM-Y][2xCOM-T][COM-Y][COM-T][COM-Y][2xCOM-U][COM-Y][COM T][9xSPACE]";
1090 fort=4to15:pokev+t,0:next
1100 fort=4to15:pokev+t,0:next
1110 pokev+21,255:pokev+16,128:pokev+46,7:poke2047,254:pokev+15,77:pokev+14,5
1120 pokev+10,70:pokev+12,210:pokev+44,0:pokev+45,0:poke2045,255:poke2046,255
1130 pokev+3,10:pokev+5,40:pokev+7,80:pokev+9,120:pokev+40,0:return
1140 d$="[HOME][20xCRSR-DOWN]"
1150 printleft$(d$,4+1)tab(1)"[CTRL-0][CTRL-7]1"
1160 printleft$(d$,11+1)tab(1)"2"
1170 printleft$(d$,16+1)tab(1)"3"
1180 printleft$(d$,7+1)tab(13)"4"
1190 printleft$(d$,7+1)tab(17)"9"
1200 printleft$(d$,10+1)tab(29)"5"
1210 printleft$(d$,13+1)tab(14)"6"
1220 printleft$(d$,13+1)tab(17)"0"
1230 printleft$(d$,14+1)tab(29)"7"
1240 printleft$(d$,18+1)tab(16)"8"
1250 return
1260 ifi4=4then1730:rem i4 = aantal sommen !!!
1270 so=int(rnd(1)*9)+1:print"[HOME][15xCRSR-RIGHT][CTRL-2]"so"[CRSR-LEFT][2xSPACE]*"t1"[CRSR-LEFT][2xSPACE]=[5xSPACE][5xCRSR-LEFT]";:i1=2
1280 poke54296,15:poke54277,7:poke54278,248:poke54273,24:poke54272,48
1290 poke54276,33:fort=1to100:next:poke54276,32
1300 so$=str$(so*t1):r1=112:r2=14:r3=48:r4=0:gosub870:pl=peek(v+30):return
1310 co=peek(1024+x2+40*(int(y1/8)+1)):i2=len(so$)+1:i3=peek(v+30)
1320 ifi1=i2and(i3and128)=128theni4=i4+1:goto1260
1330 ifi1=len(so$)+1thenreturn
1340 ifco<48orco>57thenreturn
1350 c1=val(mid$(so$,i1,1)):ifco-48<>c1thenreturn
1360 printchr$(co);:i1=i1+1:poke54296,15:poke54277,7:poke54278,133:poke54273,100
1370 poke54272,243:poke54276,33:fort=1to50:next:poke54276,32:return
1380 v=53248:print"[SHIFT-CLR][9xCRSR-DOWN][CTRL-6][COM-U][3xCOM-Y][COM-T][COM-Y][2xCOM-U][COM-Y][COM-U][COM Y][2xCOM-T][2xCOM-Y][COM-T][COM-
```

```
     Y][4xCOM-U][COM-Y][COM-U][COM-Y][2
     xCOM-T][COM-Y][2xCOM-U][COM-Y][COM
      T][COM-Y][4xCOM-U][COM-Y][3xCOM-U]"
1390 print"[4xCRSR-DOWN][CTRL-9][COM-T]
     [4xCOM-Y][COM-T][COM-Y][4xCOM-U][2
     xCOM-Y][COM-U][COM-Y][3xCOM-T][COM
      Y][COM-U][COM-Y][5xCOM-U][2xCOM-Y
     ][COM-T][COM-Y][4xCOM-T][COM-Y][5x
     COM-U]"
1400 print"[HOME][10xCRSR-DOWN][15xCRSR
     -RIGHT][CTRL-2][COM-A][10xCOM-R][C
     OM S]":printtab(15)"BBBBBBBBBBBB"
1410 printtab(15)"BBBBBBBBBBBB":printta
     b(15)"BBBBBBBBBBBB"
1420 printtab(15)"[COM-Z][10xCOM-E][COM
      X]"
1430 print"[HOME]";:fort=1to9:print"[CT
     RL 9][CTRL-6][40xSPACE]";
1440 next:print"[CTRL-0][9xCRSR-DOWN][C
     TRL 2][4xCRSR-RIGHT]je[SPACE]bent[
     SPACE]gevangen[SPACE]genomen"
1450 print"[4xCRSR-RIGHT]door[SPACE]de[
     SPACE]tafel[SPACE]monsters[SPACE]!
     !!"
1460 pokev+21,3:pokev+27,1:pokev,200:po
     kev+1,144:pokev+2,100:pokev+3,148
1470 print"[4xCRSR-RIGHT]wil[SPACE]je[S
     PACE]door[SPACE]gaan[SPACE]met[SPA
     CE]dezelfde"
1480 print"[4xCRSR-RIGHT]tafel[SPACE]of
     [SPACE]opnieuw[SPACE]beginnen??"
1490 print"[CRSR-DOWN][4xCRSR-RIGHT]jou
     [SPACE]keuze[SPACE](d/s)?":m=0
1500 geta$:ifa$="d"thenclr:fort=0to15:p
     oke53248+t,0:goto570
1510 m=m+1:poke2041,mo(m)
1520 ifm=10thenm=0
1530 ifa$="s"thenrun
1540 fort=1to50:next:goto1500
1550 print"[SHIFT-CLR][5xCRSR-DOWN][COM
      7]"tab(15)".[3xSPACE].[2xSPACE]."
     :printtab(15)"[SPACE].[SPACE].[2xS
     PACE]."
1560 printtab(15)"[3xSPACE].[SPACE].[2x
     SPACE].":printtab(15)"[SPACE].[2xS
     PACE].[2xSPACE]."
1570 printtab(15)"[3xSPACE].[2xSPACE]."
     :printtab(15)"[4xSPACE].[SPACE].":
     printtab(15)"[4xSPACE].."
1580 print"[CTRL-6]XX[CRSR-UP]X[CRSR-DO
     WN]X[COM-6]XX[CRSR-DOWN]X[CRSR-DOW
     N]X[CTRL-6]XX[2xSPACE]XXX[CRSR-DOW
     N]XXXXX[2xSPACE]X[CRSR-UP][SPACE]X
     XXX[CRSR-UP]X[CRSR-UP]X[CRSR-UP]X[
     COM-6]XXXXXX"
1590 print"[CRSR-DOWN][CTRL-1][CTRL-9][
     2xSPACE][CRSR-UP][SPACE][CRSR-DOWN
     ][CRSR-LEFT][4xSPACE]":print"[CTRL
      9][7xSPACE]":print"[CTRL-9][16xSP
     ACE]"
1600 print"[CTRL-9][40xSPACE]"
1610 print"[3xCRSR-UP]"tab(23)"[CTRL-9]
     [17xSPACE]":print"[3xCRSR-UP]"tab(
     26)"[CTRL-9][14xSPACE]"
1620 print"[3xCRSR-UP]"tab(29)"[CTRL-9]
     [11xSPACE]":print"[3xCRSR-UP]"tab(
     30)"[CTRL-9][12xSPACE]"
1630 v=53248:pokev+21,1:poke2040,245:po
     kev,180:pokev+1,150
1640 print"[4xCRSR-DOWN][CTRL-2][4xCRSR
     -RIGHT]je[SPACE]bent[SPACE]in[SPAC
     E]het[SPACE]water[SPACE]gevallen."
1650 print"[4xCRSR-RIGHT]je[SPACE]ligt[
     SPACE]nu[SPACE]op[SPACE]de[SPACE]b
     odem[SPACE]van[SPACE]de[SPACE]zee"
1660 print"[4xCRSR-RIGHT]wil[SPACE]je[S
     PACE]doorgaan[SPACE]met[SPACE]de[S
     PACE]zelfde"
1670 print"[4xCRSR-RIGHT]tafel[SPACE]of
     [SPACE]opnieuw[SPACE]beginnen??"
1680 print"[CRSR-DOWN][4xCRSR-RIGHT]je[
     SPACE]keuze[SPACE](d/s)[HOME]"
1690 print"[HOME][CRSR-DOWN][COM-7][2xC
     OM Y]EECCFCE[2xCOM-Y]EEDDCFRCD[3xC
     OM Y]EDDFDE[2xCOM-Y]EEDCFFFCD"
1700 geta$:ifa$="d"thenclr:fort=0to15:p
     oke53248+t,0:goto570
1710 ifa$="s"thenrun
1720 goto1700
1730 poke54296,15:poke54277,9:poke54278
     ,0:poke54273,50:poke54272,60:poke5
     4276,33
1740 pokev+21,129:pokev+16,129:pokev,50
     :pokev+1,115:fort=1to600:next
1750 poke54276,32
1760 poke2040,248:fort=50to100:pokev,t:
     fork=1to50:next:next
1770 pokev+21,0:print"[SHIFT-CLR][CTRL-
     1][10xSPACE][COM-A][3xSHIFT-*][COM
      S][5xSPACE][COM-A][17xSHIFT-*][CO
     M S]"
1780 pokev+21,0:print"[SPACE]seconden[S
     PACE]B[3xSPACE]B[SPACE]som[SPACE]B
     [17xSPACE]B"
1790 pokev+21,0:print"[10xSPACE][COM-Z]
     [3xSHIFT-*][COM-X][5xSPACE][COM-Z]
     [17xSHIFT-*][COM-X]"
1800 print"[2xCRSR-DOWN][CTRL-6][COM-T]
     [COM-Y][2xCOM-U][COM-Y][COM-U][2xC
     OM Y][COM-T][COM-Y][5xCOM-U][COM-Y
     ][2xCOM-U][COM-Y][2xCOM-T][2xCOM-Y
     ][COM-U][2xCOM-Y][COM-T][COM-Y][4x
     COM-U][COM-Y][2xCOM-U][COM-Y][COM-
     T][COM-Y][COM-U][COM-Y]"
1810 printtab(25)"[CRSR-UP][CTRL-8][CTR
     L 9][COM-A][8xSHIFT-*][COM-S]":pri
     nttab(25)"[CTRL-9]B[8xSPACE]B"
1820 printtab(25)"[CTRL-9]B[2xSPACE]B[3
     xSPACE]Q[SPACE]B":printtab(25)"[CT
     RL 9]B[SPACE][SHIFT-*]W[SHIFT-*][S
     PACE]Q[2xSPACE]B"
1830 printtab(25)"[CTRL-9]B[2xSPACE]B[3
     xSPACE]Q[SPACE]B":printtab(25)"[CT
     RL 9]B[8xSPACE]B"
1840 printtab(25)"[CTRL-9]B[8xSPACE]B":
     printtab(25)"[CTRL-9][COM-Z][8xSHI
     FT *][COM-X]"
1850 printtab(23)"[CTRL-6][COM-T][COM-Y
     ][2xCOM-U][COM-Y][COM-U][COM-Y][CO
     M T][2xCOM-Y][4xCOM-U][COM-Y][COM-
     U][COM-Y]"
1860 print"[COM-7][COM-P][COM-O][COM-P]
     [COM-O][COM-I][3xCOM-O][COM-I][COM
      O][COM-I][COM-O][2xCOM-P][COM-O][
     COM-P][COM-O][2xCOM-I][COM-O][2xCO
     M P][3xCOM-`][COM-P][COM-O][COM-P]
```

```
     [COM-O][COM-I][COM-O][COM-P][COM-O
     ][3xCOM-I][COM-O][COM-P][COM-O][CO
     M I]";
1870 fort=1to3
1880 print"[CTRL-9][40xSPACE]";
1890 next
1900 x=180:y=148:pokev+21,65:poke2040,2
     45:ti$="000000":pokev+16,0
1910 s=int(rnd(1)*3)+5:print"[HOME][CRS
     R-DOWN][21xCRSR-RIGHT]"t1"[CRSR-LE
     FT][2xSPACE]*"s"[CRSR-LEFT][2xSPAC
     E]="
1920 so$=str$(t1*s):so=2
1930 pokev,x:pokev+1,y-5:pokev+12,x:pok
     ev+13,y
1940 print"[HOME][CRSR-DOWN][12xCRSR-RI
     GHT]"right$(ti$,2)
1950 y=y+.1:ifval(ti$)=>14thenlo=2040:g
     osub900
1960 geta$:ifa$<>""thengosub1980
1970 goto1930
1980 ifa$=mid$(so$,so,1)thengoto2000
1990 return
2000 so=so+1:print"[HOME][CRSR-DOWN]"ta
     b(31+so)a$
2010 iflen(so$)=so-1then2030
2020 return
2030 pokev+21,0:pokev+28,1:pokev+16,0:f
     ort=40to46:pokev+t,7:next
2040 fort=2to15:pokev+t,0:next:fort=204
     1to2047:poket,254:next
2050 pokev+2,230:pokev+3,135:pokev+4,25
     4:pokev+5,135
2060 pokev+16,192:pokev+12,23:pokev+13,
     114:pokev+14,23:pokev+15,135
2070 pokev+6,230:pokev+7,114:pokev+8,25
     4:pokev+9,114
2080 pokev+21,255:print"[HOME][6xCRSR-D
     OWN][CTRL-8][25xCRSR-RIGHT][CTRL-9
     ][COM-D][9xCOM-I][COM-F]"
2090 fort=1to6:printtab(25)"[COM-K][9xS
     PACE][CTRL-9][COM-K]":next
2100 printtab(25)"[CTRL-9][COM-C][CTRL-
     0][9xCOM-I][CTRL-9][COM-V]"
2110 pokev,202:pokev+1,142
2120 print"[HOME][21xCRSR-DOWN][CTRL-2]
     [2xSPACE]je[SPACE]hebt[SPACE]de[SP
     ACE]tafel[SPACE]monsters[SPACE]ver
     slagen[SPACE]!"
2130 print"[2xSPACE]ik[SPACE]denk[SPACE
     ]dat[SPACE]je[SPACE]de[SPACE]volge
     nde[SPACE]tafel[2xSPACE]"
2140 print"[2xSPACE]nu[SPACE]wel[SPACE]
     kan[SPACE]proberen!![2xSPACE](s/d)
     ":poke2040,245
2150 geta$:ifa$="s"thenrun
2160 ifa$="d"thenpokev+21,0:goto390
2170 goto2150
2180 rem woebie
2190 data13,245,0,0,0,0,0,0,0,0,0,0,0,0
     ,0,0,0,6,1,128,15,3,192,25,134,96
2200 data0,204,0,3,255,0,3,255,0,15,255
     ,192,13,117,192,13,185,192,13,185,
     192
2210 data13,117,192,15,255,192,3,255,0,
     3,255,0,10,170,128,0,0,0
2220 data246,0,0,0,0,0,0,0,0,1,1,0,0,
     204,0,0,68,0,0,204,0,0,68,0,0,204,0
2230 data3,255,0,3,255,0,15,255,192,15,
     255,192,15,255,192,15,255,192,13,1
     85,192
2240 data15,255,192,3,255,0,3,255,0,10,
     170,128,0,0,0
2250 data247,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,1,128,0,3,192,0,6,96,0,0,48,0
2260 data3,255,0,3,255,0,15,255,192,7,2
     55,192,11,255,192,11,255,192,7,255
     ,192
2270 data15,255,192,3,255,0,3,255,0,10,
     170,0,0,0,0
2280 data248,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,0,6,0,0,15,0,0,25,128,0,48,0,3
     ,255,0
2290 data3,255,0,15,255,192,15,255,64,1
     5,255,128,15,255,128,15,255,64
2300 data15,255,192,3,255,0,3,255,0,2,1
     70,128,0,0,0
2310 rem monsters
2320 data249,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,4,0,64,5,1,64,1,69,0,0,170,0,2
     ,170,0
2330 data2,170,0,10,118,128,9,117,128,9
     ,169,128,10,170,128,10,170,128
2340 data2,170,0,2,170,0,0,204,0,10,2,1
     28,0,0,0
2350 data250,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,4,0,64,5,1,64,1,69,0,0,170,0,2
     ,170,0
2360 data2,170,0,10,214,128,9,213,128,9
     ,169,128,10,170,128,10,170,128
2370 data2,170,0,2,170,0,0,204,0,10,2,1
     28,0,0,0
2380 data251,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,4,0,64,5,1,64,1,69,0,0,170,0,2
     ,170,0
2390 data2,170,0,10,86,128,11,85,128,11
     ,169,128,10,170,128,10,170,128
2400 data2,170,0,2,170,0,0,204,0,10,2,1
     28,0,0,0
2410 data252,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,4,0,64,5,1,64,1,69,0,0,170,0,2
     ,170,0
2420 data2,170,0,10,86,128,9,87,128,9,1
     71,128,10,170,128,10,170,128
2430 data2,170,0,2,170,0,0,204,0,2,138,
     0,0,0,0
2440 data253,0,0,0,0,0,0,0,0,0,0,0,0,0,
     0,0,4,0,64,5,1,64,1,69,0,0,170,0,2
     ,170,0
2450 data2,170,0,10,94,128,9,93,128,9,1
     69,128,10,170,128,10,170,128
2460 data2,170,0,2,170,0,0,204,0,2,138,
     0,0,0,0
2470 rem data geld zak
2480 data254,0,252,0,30,253,224,31,123,
     224,31,123,224,7,119,192,3,255,128
2490 data2,0,128,1,255,0,7,255,192,15,2
     06,224,31,130,240,31,63,112,63,135
     ,88
2500 data63,243,104,63,7,104,63,206,216
     ,31,254,204,31,249,240,15,255,224
2510 data7,255,192,1,255,0
2520 rem lift
2530 data255,,,,,,,,,,,,,,,,,,,,,,,,,,,,
     ,,,,,,,,,,,,,,,,,,,255,255,255
2540 data54,71,248,11,246,64,6,153,0,0,
```

```
      0,0,0,0,0
2550  rem plons
2560  data243,0,0,0,0,3,0,141,9,136,32,1
      68,96,134,1,10,41,82,192,17,4,32
2570  data0,140,10,0,136,0,32,172,4,64,1
      72,0,0,164,0,0,132,0,0,148,0,0,180,0
2580  data0,148,0,129,182,0,1,70,2,21,73
      ,32,128,58,144,9,36,64
2590  data244,,,,,,,,,,,,,,,,,,,,,,,,,,,
      ,,,,,,,,,32,0,0,128,32,0,10,129,64
2600  data0,16,0,17,76,136,1,74,0,138,20
      1,65,64,0,8,18,34,32,0
2610  rem monster opbouw
2620  data249,250,251,251,250,249,253,25
      2,252,253
```



EINDE LISTING tafels

Checksum Tafels

REGEL 100	44	REGEL 640	66	REGEL 1180	0	REGEL 1720	81	REGEL 2260	161
REGEL 110	182	REGEL 650	54	REGEL 1190	9	REGEL 1730	226	REGEL 2270	64
REGEL 120	159	REGEL 660	95	REGEL 1200	50	REGEL 1740	39	REGEL 2280	114
REGEL 130	155	REGEL 670	50	REGEL 1210	48	REGEL 1750	48	REGEL 2290	30
REGEL 140	143	REGEL 680	212	REGEL 1220	45	REGEL 1760	147	REGEL 2300	124
REGEL 150	224	REGEL 690	93	REGEL 1230	56	REGEL 1770	76	REGEL 2310	10
REGEL 160	123	REGEL 700	101	REGEL 1240	57	REGEL 1780	179	REGEL 2320	122
REGEL 170	254	REGEL 710	195	REGEL 1250	142	REGEL 1790	65	REGEL 2330	8
REGEL 180	165	REGEL 720	242	REGEL 1260	198	REGEL 1800	79	REGEL 2340	147
REGEL 190	81	REGEL 730	114	REGEL 1270	109	REGEL 1810	143	REGEL 2350	114
REGEL 200	25	REGEL 740	46	REGEL 1280	234	REGEL 1820	65	REGEL 2360	2
REGEL 210	30	REGEL 750	155	REGEL 1290	126	REGEL 1830	25	REGEL 2370	147
REGEL 220	114	REGEL 760	169	REGEL 1300	200	REGEL 1840	108	REGEL 2380	115
REGEL 230	128	REGEL 770	57	REGEL 1310	167	REGEL 1850	51	REGEL 2390	2
REGEL 240	180	REGEL 780	77	REGEL 1320	255	REGEL 1860	7	REGEL 2400	147
REGEL 250	67	REGEL 790	146	REGEL 1330	161	REGEL 1870	143	REGEL 2410	116
REGEL 260	213	REGEL 800	23	REGEL 1340	208	REGEL 1880	42	REGEL 2420	171
REGEL 270	209	REGEL 810	46	REGEL 1350	155	REGEL 1890	130	REGEL 2430	99
REGEL 280	120	REGEL 820	51	REGEL 1360	20	REGEL 1900	83	REGEL 2440	117
REGEL 290	240	REGEL 830	99	REGEL 1370	180	REGEL 1910	163	REGEL 2450	174
REGEL 300	142	REGEL 840	49	REGEL 1380	141	REGEL 1920	209	REGEL 2460	99
REGEL 310	2	REGEL 850	217	REGEL 1390	19	REGEL 1930	74	REGEL 2470	171
REGEL 320	200	REGEL 860	138	REGEL 1400	5	REGEL 1940	150	REGEL 2480	192
REGEL 330	200	REGEL 870	25	REGEL 1410	136	REGEL 1950	235	REGEL 2490	83
REGEL 340	44	REGEL 880	188	REGEL 1420	99	REGEL 1960	222	REGEL 2500	157
REGEL 350	159	REGEL 890	142	REGEL 1430	124	REGEL 1970	86	REGEL 2510	203
REGEL 360	81	REGEL 900	64	REGEL 1440	73	REGEL 1980	160	REGEL 2520	190
REGEL 370	18	REGEL 910	249	REGEL 1450	88	REGEL 1990	142	REGEL 2530	51
REGEL 380	142	REGEL 920	222	REGEL 1460	41	REGEL 2000	237	REGEL 2540	225
REGEL 390	49	REGEL 930	243	REGEL 1470	64	REGEL 2010	1	REGEL 2550	27
REGEL 400	82	REGEL 940	227	REGEL 1480	61	REGEL 2020	142	REGEL 2560	77
REGEL 410	121	REGEL 950	142	REGEL 1490	147	REGEL 2030	249	REGEL 2570	229
REGEL 420	86	REGEL 960	129	REGEL 1500	56	REGEL 2040	161	REGEL 2580	124
REGEL 430	182	REGEL 970	215	REGEL 1510	37	REGEL 2050	234	REGEL 2590	9
REGEL 440	128	REGEL 980	123	REGEL 1520	193	REGEL 2060	69	REGEL 2600	110
REGEL 450	173	REGEL 990	193	REGEL 1530	106	REGEL 2070	244	REGEL 2610	147
REGEL 460	85	REGEL 1000	62	REGEL 1540	6	REGEL 2080	240	REGEL 2620	17
REGEL 470	213	REGEL 1010	226	REGEL 1550	50	REGEL 2090	236		
REGEL 480	62	REGEL 1020	193	REGEL 1560	108	REGEL 2100	242		
REGEL 490	137	REGEL 1030	235	REGEL 1570	181	REGEL 2110	114		
REGEL 500	142	REGEL 1040	70	REGEL 1580	100	REGEL 2120	68		
REGEL 510	143	REGEL 1050	127	REGEL 1590	33	REGEL 2130	68		
REGEL 520	142	REGEL 1060	45	REGEL 1600	239	REGEL 2140	86		
REGEL 530	47	REGEL 1070	156	REGEL 1610	107	REGEL 2150	170		
REGEL 540	97	REGEL 1080	203	REGEL 1620	108	REGEL 2160	134		
REGEL 550	253	REGEL 1090	2	REGEL 1630	173	REGEL 2170	81		
REGEL 560	72	REGEL 1100	2	REGEL 1640	201	REGEL 2180	74		
REGEL 570	118	REGEL 1110	44	REGEL 1650	148	REGEL 2190	67		
REGEL 580	167	REGEL 1120	114	REGEL 1660	64	REGEL 2200	90		
REGEL 590	104	REGEL 1130	171	REGEL 1670	61	REGEL 2210	200		
REGEL 600	7	REGEL 1140	197	REGEL 1680	159	REGEL 2220	131		
REGEL 610	215	REGEL 1150	120	REGEL 1690	226	REGEL 2230	7		
REGEL 620	165	REGEL 1160	246	REGEL 1700	56	REGEL 2240	171		
REGEL 630	29	REGEL 1170	252	REGEL 1710	106	REGEL 2250	48		

Programma Ufo

Y.S.Kasmin is voor onze regelmatige lezer een bekende inzender. Nu heeft hij weer wat schoons gemaakt. Een ruimtespel, het spel bestaat uit 5 levels. De speler begint met 5 ruimteschepen. Indien je tegen een UFO botst, verliest je één kostbaar ruimteschip. Elke neergeschoten UFO levert 25 punten op. De speler kan als hij even pauzeren wil dit doen door op run/stop te drukken. Hervatten kan men het spel ook weer door op de spatiebalk te drukken.

```
100   rem *****************************
105   rem **        ufo strike        **
110   rem **                           **
115   rem **    by yerrel kasmin      **
120   rem **                           **
125   rem ** (c) 1989 commodore-info **
130   rem ** ********************** **
135   rem         !!!! attentie !!!!
140   rem
145   rem
150   rem   voordat u deze listing gaat
155   rem   overtypen moet u eerst de
160   rem   volgende commando's invoeren
      .
165   rem
170   rem
175   rem    poke 43,1 [return]
180   rem    poke 44,64 [return]
185   rem    poke 16384,0[return]
190   rem    clr [return]
195   print chr$(147)chr$(5)
200   print"18[SPACE]sec.[SPACE]geduld[S
      PACE]a.u.b......"
205   forx=2049to4453:reada:pokex,a:nextx
210   print:print
215   print"programma[SPACE]opslaan[SPAC
      E]op:":print
220   print"1[SPACE]-[SPACE]tape":print"
      2[SPACE]-[SPACE]diskette"
225   print:print"toets[SPACE]uw[SPACE]k
      euze[SPACE](1/2)."
230   geta$:on-((a$="1")+2*(a$="2")) got
      o235,240:goto230
235   print:print"stop[SPACE]een[SPACE]c
      assette[SPACE]in[SPACE]de[SPACE]da
      tarecorder.":de=1:goto245
240   print:print"stop[SPACE]een[SPACE]d
      iskette[SPACE]in[SPACE]de[SPACE]di
      skdrive.":de=8
245   print:print"druk[SPACE]daarna[SPAC
      E]op[SPACE]return.."
250   geta$:ifa$<>chr$(13)then250
255   sys57812"ufo[SPACE]strike",de,1:re
      m file naam
260   poke193,1:poke194,8:rem startadres
       $0801 (2049 dec.)
265   poke174,102:poke175,17:rem eindadr
      es $1165 (4453 dec.)
270   sys62957:rem save
275   print:print:print
280   print"oke,[SPACE]programma[SPACE]l
      aden[SPACE]met:":print
285   print"load"chr$(34)"ufo[SPACE]stri
      ke"chr$(34)",8":print:print
290   print"runnen[SPACE]met[SPACE]'run'
      +return.":end
295   rem
300   rem
305   data 11,8,1,0,158,50,48,54,52,0,0,
      0,0,0,0,234
310   data 169,8,32,210,255,120,169,51,1
      33,1,169,208,133,3,169,48
315   data 133,5,160,0,132,2,132,4,162,3
      2,177,2,145,4,200,208
320   data 249,230,3,230,5,202,208,242,1
      69,55,133,1,169,29,141,24
325   data 208,88,169,0,162,24,157,0,212
      ,202,16,250,162,191,157,64
330   data 3,202,16,250,141,1,203,141,2,
      203,169,15,141,24,212,162
335   data 7,189,121,9,157,0,48,202,16,2
      47,76,213,8,162,0,138
340   data 157,0,4,157,0,5,157,0,6,157,0
      ,7,72,169,8,157
345   data 0,216,157,0,217,157,0,218,157
      ,0,219,104,232,208,225,96
350   data 169,42,141,72,3,141,75,3,141,
      117,3,141,120,3,162,63
355   data 189,57,9,188,249,8,157,128,3,
      152,157,192,3,202,16,240
360   data 169,14,141,248,7,169,13,141,2
      49,7,162,6,169,15,157,249
365   data 7,202,208,250,169,255,141,28,
      208,169,7,141,39,208,169,14
370   data 141,37,208,96,120,169,130,141
      ,20,3,169,9,141,21,3,169
375   data 101,141,24,3,169,17,141,25,3,
      88,32,68,229,169,0,141
380   data 32,208,141,33,208,76,185,9,0,
      40,0,0,170,0,2,190
385   data 128,130,255,130,162,255,138,4
      2,255,168,11,255,224,43,235,232
390   data 47,170,248,170,190,170,149,12
      5,86,170,190,170,47,170,248,43
395   data 235,232,11,255,224,42,255,168
      ,162,255,138,130,255,130,2,190
400   data 128,0,170,0,0,40,0,0,0,0,0,16
      0,0,0,170,170
405   data 0,149,170,0,170,128,0,42,160,
      0,10,160,0,41,106,128
410   data 170,170,160,234,165,168,170,1
      69,170,234,165,168,170,170,160,41
415   data 106,128,10,160,0,42,160,0,170
      ,128,0,149,170,0,170,170
420   data 0,160,0,0,0,0,0,0,126,251,122
      ,251,122,195,126,0
425   data 234,32,15,17,165,203,201,63,2
      08,9,169,1,133,252,169,0
430   data 141,4,212,165,203,201,60,208,
      4,169,0,133,252,162,0,30
435   data 0,48,144,8,189,0,48,9,1,157,0
      ,48,232,224,7,208
440   data 238,238,192,2,76,49,234,234,3
      2,101,14,169,11,141,17,208
445   data 32,110,8,32,246,10,169,53,141
      ,2,6,169,49,141,242,6
450   data 169,1,141,0,204,32,145,8,162,
      28,160,150,142,0,208,140
455   data 1,208,169,27,141,17,208,173,0
      ,204,201,1,208,6,32,228
460   data 11,76,31,10,173,0,204,201,2,2
      08,6,32,255,11,76,31
```

```
465    data 10,173,0,204,201,3,208,6,32,2
       8,12,76,31,10,173,0
470    data 204,201,4,208,6,32,59,12,76,3
       1,10,32,59,12,173,242
475    data 6,201,49,208,6,32,110,12,76,8
       6,10,173,242,6,201,50
480    data 208,6,32,121,12,76,86,10,173,
       242,6,201,51,208,6,32
485    data 140,12,76,86,10,173,242,6,201
       ,52,208,6,32,159,12,76
490    data 86,10,32,170,12,169,253,141,2
       1,208,169,15,141,24,212,32
495    data 187,12,165,2,201,2,208,3,76,2
       32,9,169,1,141,21,208
500    data 169,0,141,4,212,32,227,10,160
       ,0,173,27,212,41,63,170
505    data 169,0,157,128,3,162,70,32,179
       ,238,202,208,250,200,192,50
510    data 208,232,206,2,6,173,2,6,201,4
       8,240,42,162,0,160,125
515    data 142,0,208,140,1,208,32,145,8,
       238,0,208,162,205,32,179
520    data 238,202,208,250,238,0,208,173
       ,0,208,201,28,208,235,169,1
525    data 141,0,204,76,232,9,169,0,141,
       21,208,169,166,160,11,32
530    data 30,171,165,203,201,64,208,7,1
       73,0,220,41,16,208,243,76
535    data 185,9,173,24,212,9,128,141,24
       ,212,169,255,141,15,212,169
540    data 128,141,18,212,96,162,2,160,3
       1,24,32,10,229,169,136,160
545    data 11,32,30,171,162,3,134,2,166,
       2,160,31,24,32,10,229
550    data 134,2,169,146,160,11,32,30,17
       1,166,2,232,134,2,224,22
555    data 144,230,162,22,160,31,24,32,1
       0,229,169,156,160,11,32,30
560    data 171,162,4,160,32,24,32,10,229
       ,169,113,160,11,32,30,171
565    data 162,10,160,32,24,32,10,229,16
       9,121,160,11,32,30,171,162
570    data 16,160,32,24,32,10,229,169,12
       8,160,11,32,30,171,162,5
575    data 169,48,157,16,5,72,169,7,157,
       16,217,104,202,16,243,96
580    data 5,83,67,79,82,69,58,0,5,83,72
       ,73,80,58,0,5
585    data 76,69,86,69,76,58,0,158,176,1
       92,192,192,192,192,192,174
590    data 0,158,221,32,32,32,32,32,32,2
       21,0,158,173,192,192,192
595    data 192,192,192,189,0,19,29,29,29
       ,29,29,29,29,29,29
600    data 29,29,29,29,17,17,17,17,17,17
       ,17,17,5,71,65,77
605    data 69,32,79,86,69,82,13,13,13,29
       ,29,29,29,29,29,29
610    data 29,29,29,29,29,80,82,69,83,83
       ,32,65,78,89,32,75
615    data 69,89,0,169,70,162,0,157,4,20
       8,157,5,208,133,2,24
620    data 165,2,105,30,232,232,224,12,2
       08,237,238,0,204,96,169,70
625    data 162,0,157,5,208,133,2,169,240
       ,157,4,208,24,165,2,105
630    data 30,232,232,224,12,208,235,238
       ,0,204,96,169,70,162,0,157
635    data 5,208,133,2,169,100,157,4,208
       ,24,165,2,105,30,133,2
640    data 232,232,224,12,208,233,238,0,
       204,96,169,70,162,0,157,4
645    data 208,133,2,169,200,157,5,208,2
       4,165,2,105,30,232,232,224
650    data 12,208,235,238,0,204,173,0,20
       4,201,6,208,15,169,1,141
655    data 0,204,173,242,6,201,53,240,3,
       238,242,6,96,162,11,169
660    data 1,157,0,202,202,16,250,96,162
       ,11,169,1,157,0,202,72
665    data 169,0,157,1,202,104,202,202,1
       6,240,96,162,11,169,1,157
670    data 1,202,72,169,0,157,0,202,104,
       202,202,16,240,96,162,11
675    data 169,0,157,0,202,202,16,250,96
       ,32,227,10,162,11,173,27
680    data 212,41,1,157,0,202,202,16,245
       ,96,169,0,133,252,165,252
685    data 208,252,169,1,141,0,203,173,3
       0,208,173,0,220,41,2,208
690    data 3,238,1,208,173,0,220,41,1,20
       8,3,206,1,208,173,0
695    data 220,41,4,208,3,206,0,208,173,
       0,220,41,8,208,3,238
700    data 0,208,173,1,208,201,50,208,3,
       238,1,208,201,229,208,3
705    data 206,1,208,173,0,208,201,26,20
       8,3,238,0,208,201,100,208
710    data 3,206,0,208,162,6,32,179,238,
       202,208,250,238,38,208,173
715    data 30,208,133,251,41,1,240,5,169
       ,1,133,2,96,173,21,208
720    data 41,252,208,5,169,2,133,2,96,1
       73,0,220,41,16,208,28
725    data 173,21,208,9,2,141,21,208,173
       ,2,208,208,15,234,173,0
730    data 208,234,234,141,2,208,173,1,2
       08,141,3,208,173,2,208,240
735    data 43,24,173,2,208,105,3,141,2,2
       08,141,1,212,144,5,169
740    data 0,141,2,208,238,40,208,169,0,
       141,0,212,141,5,212,169
745    data 240,141,6,212,169,129,141,4,2
       12,76,157,13,173,21,208,41
750    data 253,141,21,208,169,0,141,4,21
       2,141,2,208,162,0,189,0
755    data 202,208,6,222,4,208,76,173,13
       ,254,4,208,189,1,202,208
760    data 6,222,5,208,76,187,13,254,5,2
       08,189,4,208,201,255,208
765    data 3,222,0,202,201,25,208,3,254,
       0,202,189,5,208,201,230
770    data 208,3,222,1,202,201,50,208,3,
       254,1,202,232,232,224,12
775    data 208,188,165,251,41,2,240,91,1
       73,21,208,141,255,207,169,4
780    data 141,254,207,169,3,13,254,207,
       141,21,208,173,254,207,45,255
785    data 207,240,21,174,192,2,232,232,
       142,0,206,173,192,2,205,0
790    data 206,208,248,173,30,208,208,12
       ,173,254,207,10,141,254,207,208
795    data 210,76,62,14,32,71,14,169,0,1
       41,2,208,173,21,208,41
800    data 253,141,21,208,173,254,207,77
       ,255,207,141,255,207,173,255,207
```

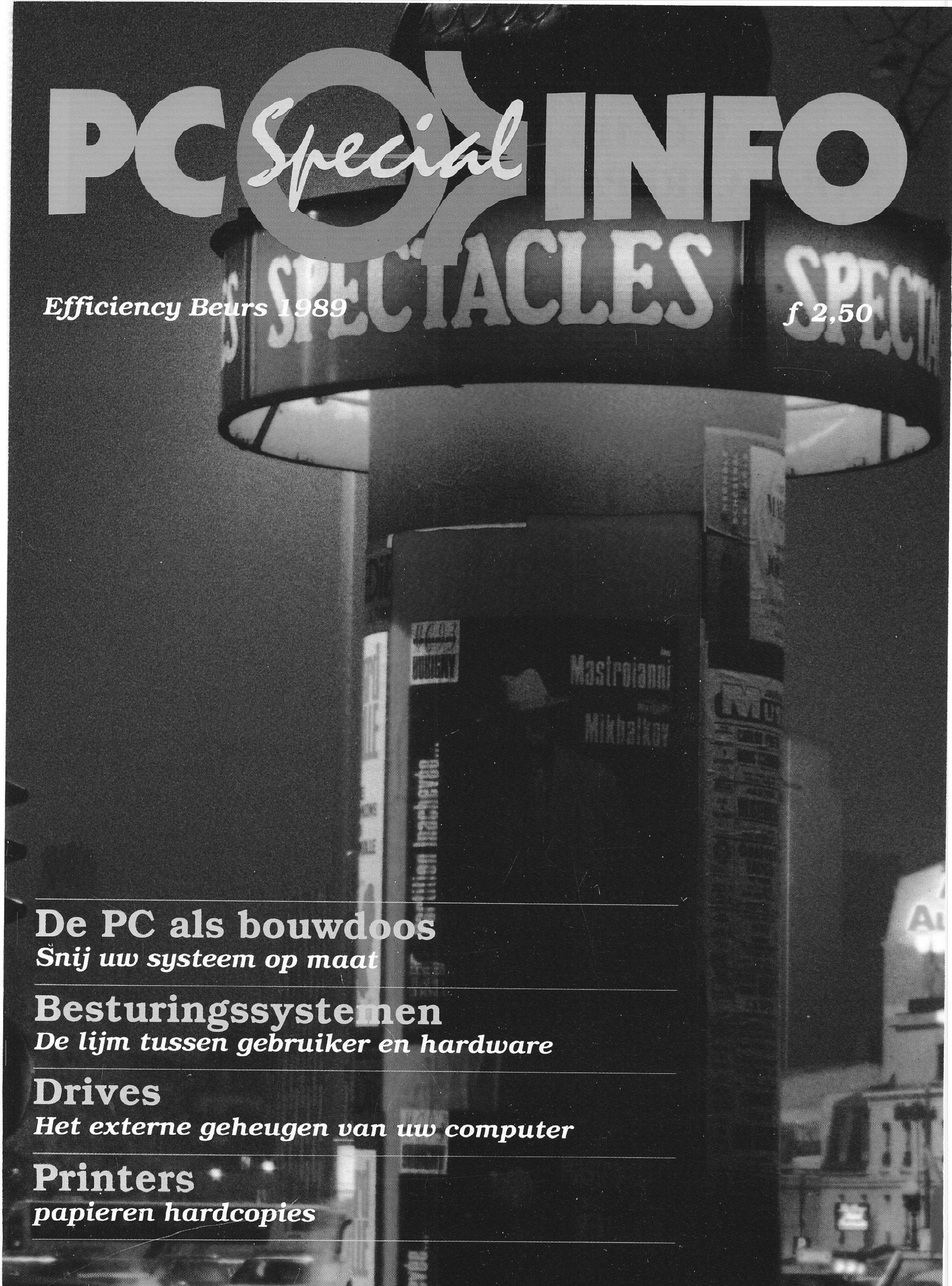
```
805    data 141,21,208,76,191,12,160,0,24
       ,162,5,254,16,5,189,16
810    data 5,201,58,208,9,169,48,157,16,
       5,202,76,76,14,200,192
815    data 25,208,229,96,169,0,141,0,207
       ,141,1,207,141,0,203,32
820    data 68,229,162,145,134,2,160,14,1
       32,3,160,0,177,2,201,43
825    data 240,11,32,210,255,230,2,208,2
       41,230,3,208,237,76,115,16
830    data 147,19,13,13,129,29,29,29,29,
       29,29,29,29,29,29,29
835    data 29,64,64,32,32,64,64,32,64,64
       ,64,64,32,64,64,64
840    data 64,13,29,29,29,29,29,29,29,29
       ,29,29,29,29,64,64
845    data 32,32,64,64,32,64,64,32,32,32
       ,64,32,32,64,13,30
850    data 29,29,29,29,29,29,29,29,29,29
       ,29,29,64,64,32,32
855    data 64,64,32,64,64,64,32,32,64,32
       ,32,64,13,29,29,29
860    data 29,29,29,29,29,29,29,29,29,64
       ,64,32,32,64,64,32
865    data 64,64,32,32,32,64,32,32,64,13
       ,156,29,29,29,29,29
870    data 29,29,29,29,29,29,29,64,64,32
       ,32,64,64,32,64,64
875    data 32,32,32,64,32,32,64,13,29,29
       ,29,29,29,29,29,29
880    data 29,29,29,29,29,64,64,64,64,32
       ,32,64,64,32,32,32
885    data 64,64,64,64,13,13,156,29,29,2
       9,29,64,64,64,64,64
890    data 32,64,64,64,64,64,64,32,64,64
       ,64,64,64,64,32,64
895    data 64,32,64,64,32,32,64,32,64,64
       ,64,64,13,29,29,29
900    data 29,64,64,32,32,32,32,32,32,64
       ,64,32,32,64,64
905    data 32,32,64,64,32,64,64,32,64,64
       ,32,64,32,32,64,64
910    data 13,30,29,29,29,29,64,64,64,64
       ,64,32,32,32,64,64
915    data 32,32,32,64,64,64,64,32,32,32
       ,64,64,32,64,64,64
920    data 32,32,32,64,64,64,13,29,29,29
       ,29,29,29,29,64,64
925    data 32,32,32,64,64,32,32,32,64,64
       ,32,64,64,32,32,64
930    data 64,32,64,64,32,64,32,32,64,64
       ,13,129,29,29,29,29
935    data 64,64,64,64,64,32,32,32,64,64
       ,32,32,32,64,64,32
940    data 32,64,64,32,64,64,32,64,64,32
       ,32,64,32,64,64,64
945    data 64,13,13,13,13,5,29,29,29,29,
       29,29,67,79,80,89
950    data 82,73,71,72,84,32,49,57,56,57
       ,32,67,79,77,77,79
955    data 68,79,82,69,32,73,78,70,79,13
       ,13,153,29,29,29,29
960    data 29,29,29,80,82,79,71,82,65,77
       ,77,69,68,32,66,89
965    data 32,89,69,82,82,69,76,32,75,65
       ,83,77,73,78,13,13
970    data 13,150,29,29,29,29,29,29,29,2
       9,29,29,29,80,82,69
975    data 83,83,32,70,73,82,69,32,84,79
       ,32,80,76,65,89,19
980    data 43,234,162,185,169,15,141,24,
       212,173,0,220,41,16,208,1
985    data 96,32,179,238,202,208,242,142
       ,4,212,142,6,212,169,42,141
```

REGEL 100	81	REGEL 295	143	REGEL 490	15	REGEL 685	158	REGEL 880	163
REGEL 105	243	REGEL 300	143	REGEL 495	27	REGEL 690	162	REGEL 885	225
REGEL 110	55	REGEL 305	10	REGEL 500	109	REGEL 695	217	REGEL 890	168
REGEL 115	104	REGEL 310	244	REGEL 505	77	REGEL 700	113	REGEL 895	160
REGEL 120	55	REGEL 315	5	REGEL 510	62	REGEL 705	204	REGEL 900	139
REGEL 125	164	REGEL 320	231	REGEL 515	172	REGEL 710	32	REGEL 905	148
REGEL 130	253	REGEL 325	31	REGEL 520	71	REGEL 715	69	REGEL 910	159
REGEL 135	245	REGEL 330	197	REGEL 525	174	REGEL 720	29	REGEL 915	148
REGEL 140	143	REGEL 335	46	REGEL 530	221	REGEL 725	120	REGEL 920	169
REGEL 145	143	REGEL 340	255	REGEL 535	86	REGEL 730	11	REGEL 925	138
REGEL 150	98	REGEL 345	230	REGEL 540	21	REGEL 735	64	REGEL 930	211
REGEL 155	241	REGEL 350	68	REGEL 545	3	REGEL 740	111	REGEL 935	148
REGEL 160	63	REGEL 355	214	REGEL 550	162	REGEL 745	20	REGEL 940	153
REGEL 165	143	REGEL 360	8	REGEL 555	3	REGEL 750	112	REGEL 945	126
REGEL 170	143	REGEL 365	97	REGEL 560	209	REGEL 755	181	REGEL 950	203
REGEL 175	26	REGEL 370	183	REGEL 565	253	REGEL 760	198	REGEL 955	240
REGEL 180	84	REGEL 375	86	REGEL 570	166	REGEL 765	99	REGEL 960	210
REGEL 185	184	REGEL 380	120	REGEL 575	210	REGEL 770	141	REGEL 965	194
REGEL 190	8	REGEL 385	22	REGEL 580	183	REGEL 775	84	REGEL 970	235
REGEL 195	154	REGEL 390	248	REGEL 585	90	REGEL 780	179	REGEL 975	189
REGEL 200	76	REGEL 395	224	REGEL 590	170	REGEL 785	208	REGEL 980	219
REGEL 205	122	REGEL 400	85	REGEL 595	85	REGEL 790	16	REGEL 985	77
REGEL 210	108	REGEL 405	252	REGEL 600	121	REGEL 795	69	REGEL 990	123
REGEL 215	61	REGEL 410	130	REGEL 605	187	REGEL 800	81	REGEL 995	195
REGEL 220	56	REGEL 415	151	REGEL 610	198	REGEL 805	80	REGEL 1000	13
REGEL 225	128	REGEL 420	89	REGEL 615	253	REGEL 810	101	REGEL 1005	146
REGEL 230	77	REGEL 425	180	REGEL 620	22	REGEL 815	115	REGEL 1010	214
REGEL 235	236	REGEL 430	154	REGEL 625	128	REGEL 820	118	REGEL 1015	100
REGEL 240	203	REGEL 435	101	REGEL 630	22	REGEL 825	4	REGEL 1020	189
REGEL 245	104	REGEL 440	33	REGEL 635	65	REGEL 830	26	REGEL 1025	75
REGEL 250	78	REGEL 445	31	REGEL 640	229	REGEL 835	164	REGEL 1030	215
REGEL 255	189	REGEL 450	161	REGEL 645	11	REGEL 840	189	REGEL 1035	107
REGEL 260	49	REGEL 455	121	REGEL 650	162	REGEL 845	130	REGEL 1040	107
REGEL 265	88	REGEL 460	213	REGEL 655	127	REGEL 850	185	REGEL 1045	188
REGEL 270	163	REGEL 465	158	REGEL 660	64	REGEL 855	155	REGEL 1050	123
REGEL 275	63	REGEL 470	21	REGEL 665	219	REGEL 860	177	REGEL 1055	219
REGEL 280	229	REGEL 475	20	REGEL 670	154	REGEL 865	207		
REGEL 285	73	REGEL 480	22	REGEL 675	169	REGEL 870	175		
REGEL 290	207	REGEL 485	82	REGEL 680	221	REGEL 875	160		

PC Special INFO

Efficiency Beurs 1989

f 2,50



De PC als bouwdoos
Snij uw systeem op maat

Besturingssystemen
De lijm tussen gebruiker en hardware

Drives
Het externe geheugen van uw computer

Printers
papieren hardcopies

Colofon

PC INFO Special

Efficiency Beurs 1989

Uitgave:
Sala Communications

Uitgever:
V.Sharfman

Redactie:
Ir L. Sala **hoofdredacteur**
drs. J. Boers **eindredacteur**
Drs. U. Schuurmans, H. Smeenk, K. v.d. Vlies, W. Scheer.

Produktie:
drs. M. de Rooij, J. Broekhuizen

Redactiesecretariaat:
R. van Zalingen
Postbus 43048
1009 ZA Amsterdam
tel. 020-246586

Illustraties:
Ben van Mierlo

Advertentie-exploitatie:
Ing. V. Sala, Ing. B. Sala,
D. van Vlijmen
Weesperstraat 103
1018 VN Amsterdam
tel. 020-273198

Administratie:
Nicole Balke & Marjo Jansen
Postbus 43048
1009 ZA Amsterdam
tel.:020-248006

PC Business INFO EB Special is een speciale uitgave rond de Efficiency Beurs 1989. Dit blad wordt in een oplage van 100.000 stuks verspreid onder de lezers van PC Business INFO, Unix Info, Commodore INFO, MSX Info en de driewekelijkse krant Computer INFO. Tevens wordt dit blad uitgedeeld onder de bezoekers van de Efficiency Beurs 1989 in de RAI te Amsterdam.

Druk: NDB, Zoeterwoude
Coverfoto: Bull Nederland

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

Inhoudsopgave

Redactioneel EB Special

Ieder jaar zijn er weer veel mensen, die voor de eerste keer met het fenomeen computer te maken krijgen. Dat dreigt, zeker bij gebeurtenissen als de Efficiency Beurs, een vergeten groep te worden. Men wil niet graag toegeven, dat al die moeilijke begrippen en dat computerjargon niet begrepen worden. Al sinds een paar jaar proberen we middels deze speciale EB uitgave juist de beginners even een opstapje te geven. Ook onder de lezers van onze reeks computerbladen zijn er ieder jaar weer vele, die zo'n algemene inleiding waarderen, daarom voegen we deze EB Special dan ook als bijlage in al onze bladen toe. Een zinvolle traditie, ook al zal de geroutineerde computeraar misschien neerkijken op het niveau. Probeer echter begrip te hebben voor hen, die nog maar aan het begin staan, en geef deze bijlage desnoods aan iemand in uw omgeving. We hopen dat er toch heel wat lezers zijn, die er wat aan hebben, al stoppen ze dit blaadje maar stiekem onderin hun bureau weg, voor het geval dat!!!

Luc Sala

MODERN MANAGEMENT

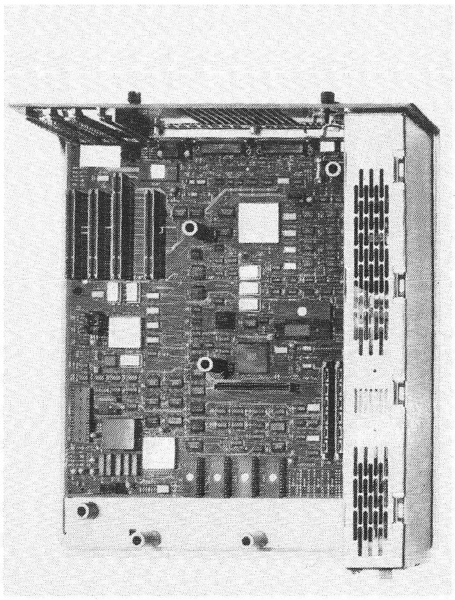
door: Berthier & Wouter Meijer



De PC is een echte bouwdoos. De systeemkast van een IBM compatible PC, Amiga 2000 of Macintosh biedt ruimte voor uitbreidingen, zoals extra interfacekaarten, meer RAM, een harde schijf of diskettestation, tapestreamer, modem-, fax- of scannerkaart. Dankzij deze open-systeem architectuur kunt u de PC precies op maat snijden.

De PC als bouwdoos

Snij uw systeem op maat



IBM PS/2 model 50
Op de systeemkaart zijn de uitbreidingsslots zichtbaar

Bij de PC valt de uitbreidingsruimte in twee delen uiteen: de **uitbreidingsslots** waarin extra elektronicakaarten gestoken kunnen worden en het **drive chassis** waarin extra diskdrives of een harde schijf gemonteerd kan worden. Enkele PC's bieden ook nog de mogelijkheid om de systeemkast modulegewijs uit te breiden. Een kastje met een extra drive of geheugen kan aan of op de moederkast geklikt worden. Dit is bijvoorbeeld het geval bij de Amiga 500 en de Tandon PC- modellen met uitwisselbare harddisk.

Redenen voor uitbreiding

In de praktijk blijkt een PC veelal met de eisen van de gebruiker mee te groeien. Zijn in eerste instantie twee diskdrives misschien voldoende, dan wordt er later toch maar een 20 MB harde schijf bijgezet. Die saaie Hercules kaart wordt vervangen door een EGA kleuren videokaart en het geheugen uitgebreid. Daarmee zijn meteen al drie belangrijke uitbreidingsredenen ter sprake gekomen:

- Meer opslagcapaciteit van de drives
- Overstap van monochroom naar kleurenvideo
- Uitbreiding van het RAM geheugen

Drie andere belangrijke redenen zijn:

- Inbouw van extra PC-functies; bijvoorbeeld een modem, fax, game adapter of digitizer
- Behoefte aan meer aansluitmogelijkheden (voor muis, modem, printer, netwerk, etc.) in de vorm van seriële en parallelle poorten of netwerkkaarten
- Versnellen van het rekenwerk met een andere CPU of coprocessor

Daarnaast zijn er nog tal van andere specifieke redenen die we verder in het midden laten.

Uitbreidingsslots

Doorgaans ligt onder in de systeemkast de **hoofd-** of **moederkaart**. Deze groen gecoate plastic kaart bevat de elektronische onderdelen (oa. de chips en microprocessoren) en hun onderlinge verbindingen. Bovenop deze moederkaart staan zwarte langwerpige 'contactdoosjes' met sleuven, waarin met een goudlaagje bedekte contacten zitten. De lange sleufvormige contactdozen worden **expansion slots**, kortweg slots genoemd.

In een slot kunt u de connector, het met contactpunten bedekte uitsteek-

sel aan de onderzijde, van één uitbreidingskaart steken. Plug de kaart erin en de computer kan een taak meer verrichten of beschikt nu over extra geheugen.

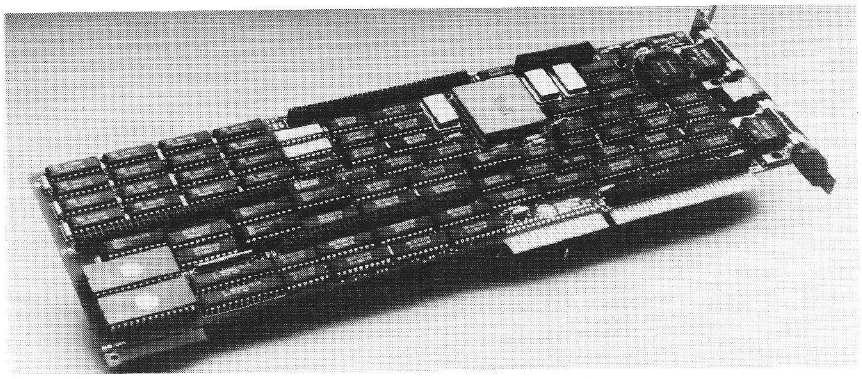
Slots zijn er in meerdere maten en soorten. Het oudste type is het **XT**-slot voor 8-bits communicatie op de IBM PC XT-bus. NB: via de bus communiceert de elektronica op de hoofdkaart met de in de slots gestoken interfacekaarten. De verbeterde **AT**-slots maken een aanzienlijk snellere 16-bits datacommunicatie mogelijk op de AT-bus. Betrekkelijk nieuw zijn de IBM **PS/2**-slots en de **EISA**- slots.

De slots stellen eisen aan de interfacekaarten. Een AT kaart past niet in een XT-, PS/2- of Amiga-slot. PS/2 past, net als de eigen Amiga-kaarten, alleen op zich zelf. En de XT kaarten passen zowel in XT- als AT-slots. In het laatste geval natuurlijk alle met XT-werking! De EISA-bus poogt meer universeel te zijn en accepteert naast de EISA- ook XT- en AT-connectoren. Het zal duidelijk zijn dat u bij de aankoop van uitbreidingskaarten hiermee rekening dient te houden.

Hoeveel vrije slots per machine beschikbaar zijn hangt van het ontwerp en de uitvoering af. Vrijwel altijd zijn er al enkele slots bezet met een videokaart, multifunctiekaart en drivecontroller. Dan houdt de gebruiker zo'n 5 tot 8 lege slots over.

Kaarten

De naam **kaart** slaat op de platte vorm van de elektronische printplaat. In feite is een uitbreidingskaart gewoon een print met daarop een aantal speciale chips plus de ondersteunende elektronica. Die chips bepalen wat de kaart voor de PC gaat doen. Zet er extra RAM-chips op en de computer krijgt de beschikking over meer geheugen. Monteer chips voor aansturing van gedetailleerde kleurenbeelden en de fraaiste graphics verschijnen op het beeldscherm. Een klokje met batterijvoeding voorziet de XT van een ingebouwde klok. En communicatiechips verzorgen de modemverbinding, de fax of het netwerk.



We geven hier een kort overzicht van de meest gebruikte uitbreidingskaarten:

- **Videokaarten** voor het uitsturen van monochrome of kleurenbeelden naar een geschikte monitor
- **Drive-Controllerkaarten** verzorgen het dataverkeer tussen de diskdrives en/of harddisk met de microprocessor (= centrale rekeneenheid) van de PC
- **Geheugenkaarten** breiden het RAM uit. Hoe meer RAM- geheugen des te meer en sneller kan de in het RAM opgeslagen informatie verwerkt worden. Van belang voor grote spreadsheets, databases, gedetailleerde videobeelden, e.d.
- **Modemkaarten** voor het communiceren met andere PC's over de telefoonlijn
- **Netwerkkaarten** voor het aansluiten van meerdere PC's op een centrale dienstverlener ('server') voor het beheer van de databank en/of gezamelijke printer en de lokale communicatie tussen de kantoormachines
- **Multifunctiekaarten** voorzien het systeem van een batterij gevoede klok, extra seriële en/of parallelle aansluitpoorten en aansluiting voor een spelpook (= game adapter)

Noviteiten EB '89

De Efficiency Beurs '89, nu voor de derde keer als jaarlijks evenement, is natuurlijk weer een enorme happening. Toch kunnen wij ons niet aan de indruk onttrekken dat de interesse van de kant van het publiek, maar ook van de kant van een aantal grote computerleveranciers, aan het afnemen is. Het wegblijven van IBM is voor de organisatie toch wel een klap, en ook bedrijven als Apple, Bull, Digital Equipment en Sun zijn afwezig. Apple heeft natuurlijk de MacWorld, Bull organiseerde begin september in de Bull(e)wijk het Forum Automatisering in bedrijf. Andere groten zijn er natuurlijk wel, zoals Hewlett Packard, Olivetti, Siemens, Commodore en Atari. De rangen van de PC leveranciers zijn verder ruim vertegenwoordigd en software en toebehoren zijn in alle soorten en maten te bewonderen. Het echte nieuws is niet dik gezaaid, maar daar zijn we de laatste jaren wel aan gewend. Let vooral op de 386's, de 486 van Apricot, grote en kleine portables en de proliferatie van grafische gebruikersinterfaces die allemaal op elkaar lijken, maar onder verschillende besturingssystemen draaien.

Amtron

D 709 03403 - 79690

De serie van LEO computersystemen is uitgebreid met een 386-25 MHZ en een 386-33 MHz tower. Beide machines hebben ook een Intel 82385 cache controller aan boord. Het cache geheugen bedraagt 64 KB. Het 4 MB geheugen is uit te breiden tot 16 MB. Een VGA videokaart met 512 KB geheugen, 5,25" diskdrive, diverse poorten e.d. zijn aanwezig. Harddisks zijn optioneel.

Ook levert Amtron twee nieuwe monitors van Princeton: de Ultra 14 en de Ultra 16, die compatible zijn met de CGA-SuperVGA standaards. De dot pitch bedraagt 0.31 mm. en de maximale resolutie resp. 800 x 600 en 1024 x 768. Een andere nieuwe monitor, de MM-411, komt van Amtronics. Deze flatscreen

Special

heeft een beelddiameter van 14" en een scanfrequentie van 18.4 KHz, de bandbreedte is 18 MHz. De monitor is Hercules en IBM compatible.

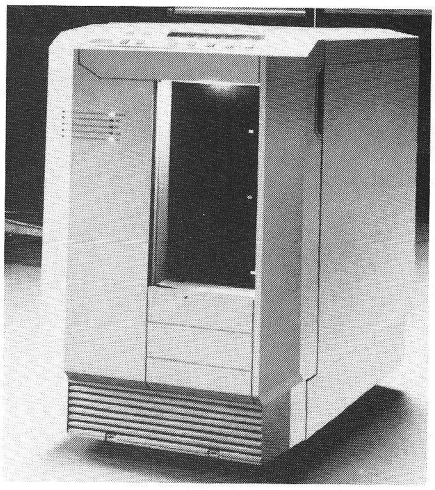
Gestetner heeft twee nieuwe laserprinters die door Amtron getoond worden. Het zijn de LP-600 en de LP-800. De 600 is HP Laserjet II compatible en de 800 is Postscript compatible. De resolutie bedraagt bij beide machines 300 x 300 dpi.

De PS-388 Postscript emulatiekaart van Princeton is Adobe Postscript compatible en heeft standaard 35 Postscript fonts. De controllerkaart past in elke PC/XT/AT en PS/2-30 compatible machine. De kaart maakt gebruik van een Weitec 10 Mips Risc chip.

Ook heeft Amtron een aantal nieuwe Ethernet en Digiboard produkten op zijn stand.

Apricot Systems

O 135 *023 - 340884*



Waarschijnlijk de eerste seriemachine met de 80486 ter wereld, de Apricot VX/FT, zal door Apricot Systems op stand 135 in de Oosthal worden getoond. De i486 heeft een geïntegreerde rekenkundige processor en is in de VX/FT gebouwd op een moederbord met de MCA busstructuur. De computer heeft een prestatieniveau van 15 keer de oorspronkelijke DEC VAX computer (15 MIPS). De aanduiding FT in de naam staat voor Fault Tolerant, wat weer slaat op de ingebouwde Uninterruptable Power Supply waarmee de machine nog maximaal twee uur door kan werken als de netstroom is uitgevallen. Daarnaast is er door speciale warmtesensoren

- Kaarten voor een **extra microprocessor** (coprocessor). Deze coprocessor kan de machine flink versnellen door de centrale processor, de **CPU**, van een aantal tijd-consumerende taken te ontlasten.
- **Digitizers** of **scanners** voor het uitlezen van tekst of grafieken/tekeningen, foto's e.d. door de PC
- **Besturingskaarten voor randapparatuur**; bijvoorbeeld een kaart voor het aansturen van een tapestreamer
- Ook een geheel **nieuwe hoofdkaart** ('mother board') behoort tot de mogelijkheden. Is het moederbord met CPU verouderd, dan kunt u dit er uit halen en door een krachtiger update vervangen. De ombouw van een XT tot AT, of AT tot 80386 machine wordt daarmee relatief eenvoudig.

Van de meest gangbare kaarten bestaan tal van uitvoeringen in verschillende prijsklassen. Neem bijvoorbeeld de Hercules videokaart. Een echte standaard voor monochrome PC-beelden, die door tientallen fabrikanten tegen concurrerende prijzen geleverd wordt. Het aanbod is groot en let bij aankoop dan ook op de kwaliteit van de onderdelen, de handleiding, de te verwachten systeemprestaties (AT of XT), het gebruiksgemak, de installatie en of de kaart compatible is met de hardware.

Kaartruimte

Het fysieke formaat van kaart en slotomgeving zijn bepalend voor het al of niet passen van de interfacekaart. In krappe systeemkasten, oa. bij compacte en draagbare modellen, gaan alleen zogenaamde **halflange kaarten**. Een groot bureau- of towermodel systeemkast kan ook **fullsize kaarten** aan.

Een ander probleem is de dikte van de kaart. Stevige kaarten, bijvoorbeeld een hardcard waarop een harde schijf gemonteerd is, kunnen qua breedte wel eens twee slots aan ruimte opslokken!

Drive chassis

Zoals u in het artikel over drives kunt lezen is meer dan één diskdrive en/of de inbouw van een harde schijf een voorwaarde om snel en comfortabel te kunnen werken. Een diskettestation of harddisk wordt voor de PC in twee vaste inbouwmaten geleverd. De breedste is de **5.25 inch versie**. De smalste de **3.5 inch versie**. Het smalle type past met geschikte adapter wel in een 5.25 inch chassis. Omgekeerd niet.

Voor drive-uitbreiding bent u naast de mogelijkheden van de **controllerkaart** afhankelijk van de ruimte in het **drive chassis**, ook wel 'drive bay' genoemd. In de grotere kantoormachines en towerkasten kunt u met gemak vier tot zes drives plaatsen. Bij compacte modellen en portables is de ruimte dikwijls met één extra drive al vol.

Voeding

Extra drives, kaarten en sommige randapparatuur stellen hoge eisen aan de **stroomvoorziening**. Kan bij een kleinere kantoor-PC 150 Watt voeding voldoende zijn, bij een volledig uitgebouwde toren-PC komt u al spoedig op een behoefte van 200 Watt of meer.

Een tweede beperking vormt het aantal beschikbare stroomkabeltjes dat aan de PC voeding ontspringt. Vier extra drives wil vaak nog wel lukken. Zes vormt een probleem.

Kastruimte

Niet elke systeemkast biedt evenveel uitbreidingsruimte. De grote **tower modellen** zijn berekend op meer en zwaardere uitbreidingen dan de doorsnee PC gebruiker kan bedenken. Ook de **normale kantoorkasten** zullen in de meeste gevallen voldoende ruimte en voeding bieden.

Geheel anders wordt het bij de **minitorens**, **compact modellen** en

portables. Daarin wordt het woekeren met de vrije slots, kaart- en driveruimte. Ook de voeding kan te krap bemeten blijken.

Blijkt de ruimte in uw PC te klein, dan zal er naar externe mogelijkheden gezocht moeten worden. Bijvoorbeeld:

- Een **expansion box**. Dit is een soort uitbreidingshuis met vrije slots en een drive chassis dat naast de moeder-PC geplaatst wordt. De moeder bestuurt deze dochter alsof de uitbreidingen in haar eigen systeemkast zitten
- **Add/click on modules** die via een uitbreidingsbus of kabeltje met de systeemkast verbonden worden. Een bekend voorbeeld is de Amiga 500 die middels modules van extra RAM en drives voorzien kan worden

Het nadeel van deze uitbreidingen is het feit dat de compacte vormgeving en de extra bureauruimte verloren gaan. Daarom is het, met de toekomstige uitbreidingen in het achterhoofd, beter om direct voor een groter model PC te kiezen dan later met ruimte te moeten woekeren of externe uitbreidingen aan te moeten sluiten.

RAM chips

Nog vaak worden PC's met slechts 512 KB aan **vrij RAM** verkocht. Steeds meer programma's vragen echter al om 640 KB. Op een **XT** kunt u veelal op de hoofdkaart uitbreiden door gelijksoortige RAM chips in de drager (sockets) te steken.

Bij de **AT, 386 en 68.000 machines** zijn uitbreidingen tot vele MegaBytes mogelijk. Stel daarbij altijd de vragen:

Welk soort Chips of uitbreidingskaart zijn nodig? Hoeveel RAM heeft de software eigenlijk nodig? Wordt het allemaal niet te duur?

Bouwen is niet zo moeilijk als u denkt

Veel PC gebruikers durven hun machine niet eens open te maken, laat staan er iets extra's in te zetten. Er zou immers snel schade aan zo'n ingewikkeld apparaat kunnen ontstaan? Gelukkig valt dat allemaal best mee. Iedereen die een schroevendraaier kan hanteren en twee stekkeruiteinden in elkaar kan steken, is in staat om zelf zijn/haar PC op maat te verbouwen.

Het openen van de systeemkast staat in elke goede handleiding en in tal van PC doe-boeken beschreven. Hetzelfde geldt voor het inpluggen van kaarten en het plaatsen van extra drives. Ook kunt u de leverancier om raad vragen en kijken hoe het bij gelijksoortige PC's zit. Neem wel even de volgende gouden regels in acht:

- Schakel altijd de stroom uit
- Voorkom statische ontladingen. Grijp eerst een metalen buis of bureau vast alvorens aan de kaarten te komen
- Gebruik passend gereedschap. In de praktijk twee schroevendraaiers
- Forceer nooit iets
- Lees de handleiding
- Let goed op de plaats van de aangegeven markeringen
- Vraag de leverancier om de bij uw systeemkasttype behorende schroefjes, geleiderails, kaartklemmen e.d.
- Blijf overal af waar u niet aan hoeft te komen
- Denk aan eventuele garantiebepalingen

Echt, het is allemaal niet veel moeilijker dan het monteren van een nieuw stopcontact.

Het bouwdoosconcept of open-systeem architectuur van de Personal Computer maakt dat u de PC geheel op maat kunt snijden. Natuurlijk blijft het verstandig om een zo compleet mogelijk systeem te kopen, maar er kunnen aanvankelijk budgetaire beperkingen zijn. Verder kunnen toekomstige uitbreidingen vaak nog niet worden voorzien. Mocht de eigenaar in de toekomst meer of nog sneller willen, dan biedt de grote PC daartoe de ruimte. Om de prijs en de installatie behoeft u het echt niet te laten.

U.S.

voor gezorgd, dat mogelijke elektronische fouten in een vroeg stadium gesignaleerd kunnen worden. Een ander Fault Tolerant aspect is de mogelijkheid van 'mirroring' van de ene harddisk op een tweede model.

De VX/FT heeft een maximale harddisk capaciteit van 5 Gigabyte en er is een DAT recorder voorzien voor backups tot 1,2 GB per tape. De machine heeft 7 vrije MCA slots. Er zijn vier modellen van de 800 waarvan het eerste model, de 800/10, beschikt over de 80386 (25 MHz). De 800 series bestaat uit de modellen die bedoeld zijn als Unix host, terwijl de 400 serie zal worden gepositioneerd als netwerkserver.

Argo

A 221 — 03465 - 95211

Met de Dancall autotelefoon bestaat de mogelijkheid om middels een Line Interface Module diverse andere apparatuur op de telefoon aan te sluiten. Er kunnen telefax, modem, telefooncentrale en dergelijke gekoppeld worden.

Ashton-Tate

Z 401 — 020 - 462515

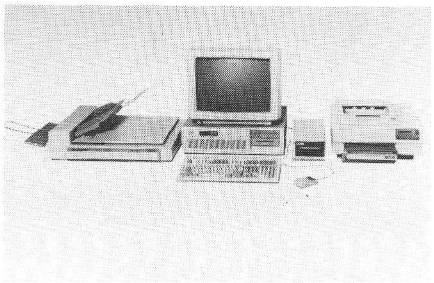
Samen met geautoriseerde dealers, zoals Positronica, Tritech, Olivetti en HP, zal Ashton-Tate haar software dBase IV, SQL Server en het kantoorautomatiseringspakket Framework III tonen. Op de eigen stand demonstreert Ashton-Tate ook de nieuwste versie van RapidFile, een file manager met geïntegreerde rapportgenerator en tekstverwerker voor bestandsverwerking, memo's en mailmerge. De nieuwe versie 1.2 van RapidFile biedt een aantal nieuwe opties voor (Nederlandse) spellingcontrole, afdrukken en gebruik van dBase en Lotus bestanden.

De Blaeuwe Werelt

E 356 — 055 - 274848

De Blaeuwe Werelt houdt zich vooral bezig met computerformulieren, druk- en verwerkingsapparatuur en het opslaan van documenten. Een primeur op de beurs is een nieuw ontwikkelde kwaliteit recycled papier, een lovenswaardig produkt

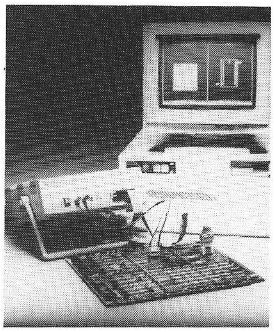
gezien de enorme stapels die elke dag weer door talloze printers gejaagd worden.
De Blaeuwe Werelt toont ook een nieuwe lijn op- en afrollers van Dalren. Deze machines voorkomen dat het printproces telkens weer gestopt moet worden omdat een nieuw pak computerformulieren ingevoerd moet worden.
De datasystems divisie van De Blaeuwe Werelt levert oa de Inform systemen voor handtekeningen verificatie en toegangscontrole.



Nieuw is het DocMan systeem voor documenten automatisering - het vastleggen van documenten, foto's, tekeningen, brieven, etc. op optische schijf. DocMan is geschikt voor gebruik op de PC.

Colijn Computers

B 528 ***01100 - 32256***
Colijn presenteert onder andere het Uni-Living software pakket, Sherry computers en de Boardwizard.



Uni-Living is speciaal ontwikkeld voor de woninginrichting branche. Het biedt alle financiële en andere administratieve modules voor grote en kleine bedrijven. De Sherry computerserie is uitgebreid met een 33 MHz 386 systeem. Met het Boardwizard meetinstrument kunnen technische storingen in printed circuit boards worden opgespoord.

PC's upgraden

PC's kunnen geen rekenkracht genoeg in huis hebben. Bezitters van Personal Computers worden voortdurend geconfronteerd met het *nog sneller* uit de advertenties. De 25 MHz versie is nog niet uit of de 33 MHz machine wordt reeds aangekondigd. Hetzelfde geldt voor CPU's, systeembussen en harddisks. Als je het allemaal moet geloven staat er drie maanden na aanschaf van een schitterende PC een prehistorisch prul op het bureau.

Advertenties en sterke verhalen over data verslindende computers dient de serieuze PC-gebruiker altijd met een fikse korrel zout te nemen. Sneller betekent niet automatisch beter en de al wat oudere PC kan nog altijd prima geschikt zijn voor de opgedragen taken. Maar stel nu dat de oude machine toch wat traag wordt? Gewoon een recenter en krachtiger exemplaar kopen is meestal slechts voor de beter gebudgetteerden weggelegd of betekent het verkennen van de tweede hands markt. Een andere mogelijkheid is het opvoeren, **'upgraden'** van de bestaande PC.

Snellere CPU

Het versnellen van de PC kan op vele manieren. Voor de hand ligt de keuze van een **snellere CPU** of het **opvoeren van de kloksnelheid** van de bestaande microprocessor. Als voorbeeld neem ik een al wat bejaarde XT met Intel 8088 CPU op 4,77 of 8 MHz. In de radiodump zijn oscillatorkristallen te koop waarmee dat 8088-beestje nog tot 10 of 12 MHz opgedreven kan worden. Dat 'opdrijven' staat er niet voor niets, want de Intel 8088 werd niet voor dergelijke snelheden ontworpen. De kans op storingen en fouten neemt navenant toe en het blijft de vraag of deze problemen een snelheidswinst van 10 tot 30% goed maken.

Een andere mogelijkheid is het vervangen van de XT-CPU voor een sneller type van bijvoorbeeld NEC (V20 voor de 8088, en V30 voor de 8086). Deze chip is goedkoop en inderdaad iets sneller. Weer blijft het de vraag of het weghalen van de chip en enkele tientjes voor de CPU de winst van 10 tot 20% wel rechtvaardigt.

Van XT naar AT

De **XT ombouwen tot een AT** lijkt een veelbelovend project. Die systeemkast, diskdrives, voeding en interfacekaarten heb je immers al. Daar behoeft alleen nog maar een Intel 8086 op de hoofdkaart gedrukt te worden. Ho ho, wacht eens eventjes. Was er bij de AT ook niet sprake van een verbeterd technisch ontwerp? Juist ja, het gaat wezenlijk om een andere machine met afwijkende ondersteunende elektronica. Om in één klap van al het gezeur af te zijn kan de gebruiker gewoon de gehele hoofdkaart vervangen en de RAM chips overzetten (er kunnen extra kosten in de aanschaf van snellere RAM chips schuilen, maar de resultaten zijn er wel naar). Dan werkt de zaak meteen goed en zit er een echte 16-bits bus in de PC. Op computerbeurzen liggen redelijke tot spotgoedkope moederkaarten voor het oprapen. Let wel even op de fysieke afmetingen. Past de kaart in de oude systeemkast?

Een andere methode is de oude 8086/88 weg te trekken, een versnellerkaart in een vrij busslot te drukken en deze met een kabeltje plus plug aan de CPU-drager op de hoofdkaart te koppelen. Op die **accelerator-kaart** zit een snellere 16-bits 80286 CPU die de benodigde rekentijd aanzienlijk verkort. Er kleven wel enkele bezwaren aan deze methode:

- Buiten de versnellerkaart om blijft het systeem de oude eigen bus voeren. Een ex-8088 PC blijft gewoon 8-bits voor de buitenwacht!

- Een **traag RAM** doet de versneller een pas op de plaats maken. De CPU kan niet sneller rekenen dan de Von Neumann bottleneck (= data-uitwisseling tussen CPU en RAM banken) toelaat.

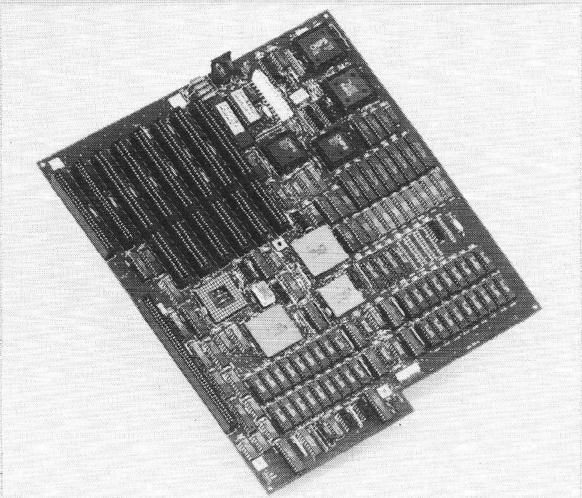
Een XT echt helemaal tot een AT verbouwen blijkt in de praktijk dikwijls een dure illusie te zijn. Zijn de CPU en het RAM vervangen, dan zijn er nog de diskdrives (360 KB in plaats van 1.2 MB), de XT drivecontroller, incompatible I/O- of multifunction kaarten en de trage XT harddisk. Tot overmaat van ramp blijkt soms ook de voeding te krap bemeten.

Dan maar helemaal niet verbouwen? Een vraag die moeilijk te beantwoorden valt. De verbouwingskosten zijn voor een budget XT-kloon bijna net zo hoog als een nieuwe AT. Als u daarentegen over een nog in goede staat zijnde merk XT beschikt, kan upgraden met een kwaliteits accelerator wel degelijk lonen. En natuurlijk is zelf knutselen ook heel leuk. Met op beurzen verkregen onderdelen kan iedereen zonder twee linkerhanden een complete PC bouwen.

Naar de 386

Inmiddels zijn er ook **80386/80386SX acceleratorkaarten** en **losse hoofdkaarten** verkrijgbaar. Hierbij gelden weer dezelfde bezwaren. De snelheidswinst kan weliswaar meer dan 200 tot 300% zijn. Daarvoor moet echter wel een stevige prijs betaald worden.

De ombouw van een goed uitgeruste AT naar een 386 model kan lonen. De prijs van een losse hoofdkaart ligt flink lager dan die van een complete machine.



33 MHz 80386 moederboard van Hauppage

Coprocessoren

Mathematische coprocessoren zijn bedoeld om de CPU van dom rekenwerk te ontlasten. Bovendien rekenen deze chips hun standaard berekeningen aanmerkelijk sneller door dan de centrale microprocessor die al zoveel aan het hoofd heeft.

De installatie van een mathematische coprocessor helpt alleen als de software er ook mee kan werken. Anders heeft die extra investering van ettelijke honderden guldens geen zin. De behaalde snelheidswinst kan bij geschikte programmatuur hoog zijn.

Andere upgrades

Er zijn diverse mogelijkheden om naast het vervangen van CPU's de PC op te voeren. Bijvoorbeeld:

- De **installatie van snellere drives**
- Het **gebruik van een sneller RAM**
- **Meer RAM** in de machine zetten
- Een **snellere drivecontroller** in de PC zetten
- Een **efficiënter besturingssysteem** nemen

Bij al deze upgrades zult u zich steeds weer moeten afvragen of de computer daarvoor geschikt is en of de kosten tegen de baten opwegen.

Een computer is eigenlijk een verzameling hardware dat hoogstens zelf naar de opstartroutine kan zoeken. Zonder een besturingssysteem doet de PC helemaal niets nuttigs. De communicatie met de gebruiker, de drives, andere computers, randapparatuur, het geheugenbeheer en indelen van bestanden zijn zonder een Operating System als MS-DOS, UNIX of OS/2 niet mogelijk. Practisch gezien vormt het besturingssysteem de 'lijm' tussen de hardware en de gebruiker.

Besturingssystemen

De lijm tussen gebruiker en hardware

In de systeemkast van een PC vindt u de microprocessor, de ondersteunende elektronica, de interface kaarten en de drives. Aan die kast hangt weer de nodige randapparatuur, zoals toetsenbord, muis, printer, modem en monitor plus de gebruiker die de PC wil bedienen. Om al die componenten nu tot één voor de gebuiker beheersbaar systeem te verenigen is een **besturingssysteem** (OS of Operating System) nodig. Het operating system vervult daarbij onder meer de volgende belangrijke functies:

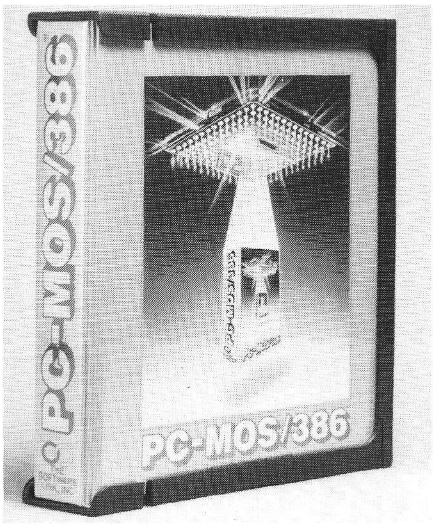


Systeem management programma's met vensters vormen een gebruiksvriendelijke interface tussen Operating System en gebruiker (foto Bull Micral 35)

- Het **correct installeren** van de hardware voor het gebruik. De OS-software leert de computer als het ware wat er aan elektronica, drives, RAM chips, kaarten en randapparatuur binnen het systeem zit en wat de PC daarmee kan doen. Dikwijls wordt het OS hierbij ondersteund door zogenaamde **drivers**, kleine hulpprogrogramma's voor het aansturen/installeren van extra geheugen, RAM disks, modems, muizen, printers e.d. Hardware, RAM disks en RAM-residente programma's die bij het opstarten geïnstalleerd moeten worden, staan in het configuratiebestand. Bij MS-DOS is dat het belangrijke file **CONFIG.SYS**.
- De **communicatie** tussen de verschillende systeemcomponenten. Het besturingssysteem wordt ook wel **Disk Operating System**, (DOS) genoemd. De term 'disk operating' geeft al aan dat één van de voornaamste OS-taken het besturen van de drives is. Al het I/O-dataverkeer staat onder controle van dit DOS. Bij systemen als UNIX, XENIX en PICK is het OS van huis uit al geschikt voor netwerktaken. Bij MS-DOS en OS/2 is extra netwerksoftware een must.
- Het **besturen van allerlei randapparatuur**

- Het **indelen van de bestanden (files) en (sub-) directories**
- Het **bewerken van bestanden**. Oa. kopiëren, wissen, verplaatsen, overschrijven, backuppen, beveiligen, enz.
- Het **formatteren** en **indexeren** van schijven.
- Het creëren van een of ander **gebruikersinterface**. Dat deel van het OS dat met de computergebruiker communiceert.
- **Boekhoudkundige taken**, zoals waar informatie zich in het geheugen of op de schijf bevindt en het bijhouden van een systeemlogboek.
- Het **beheersen van de taakverdeling en taakuitvoering** binnen het systeem. Hieronder vallen zaken als welke opdracht aan de beurt komt voor processortijd, het uitvoeren van meerdere taken tegelijk (**multi-tasking** en/of het tegelijkertijd toelaten van meerdere gebruikers op één (**multi-user**) systeem.

Behalve de hier genoemde hoofdtaken kennen de verschillende operating systemen nog tal van subtaken. Het zou te ver voeren om ze hier uitvoerig te bespreken.



PC-MOS/386 is één van de vele (minder bekende) besturingssystemen. Het is een multi-user en multi-tasking systeem voor 386 machines

Hoe zit een OS in elkaar?

Een besturingssysteem wordt meestal onderverdeeld in het Kernel, de Shell, de Utilities en Gebruikersprogramma's. Het **Kernel** vormt de kern van het OS die het systeem bestuurt op het niveau van de laagste hardwarefuncties. In feite creëert het Kernel de werkomgeving op hardwareniveau voor de daarboven liggende hogere delen van het OS. Die hogere delen 'vragen' het Kernel om bepaalde hardware taken voor hen uit te voeren. Tot de Kernel- taken behoren oa. het regelen van de input/output (I/O-functies), het geheugengebruik, drivebesturing, timing, multi-user- en multi-tasking operaties. Verder houdt het kernel meestal ook het systeemlogboek en (een deel van) de boekhouding bij.

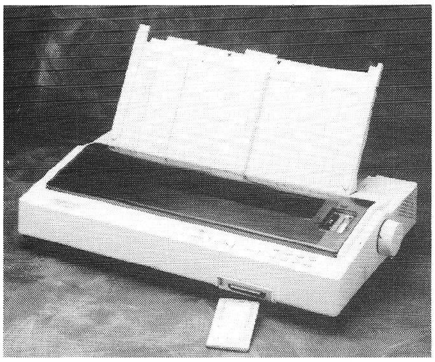
Daar het Kernel de werkomgeving voor de andere lagen van het OS schept en primair op het laagste hardwareniveau werkt, dient het exact voor de CPU en diens ondersteunende hardware geschreven te zijn. Met name de I/O-drivers voor de verschillende devices luisteren zeer nauw. Is het besturingssysteem flexibel, dan kan het OS door aanpassing van het Kernel op vele verschillende computertypen draaien. Daarmee wordt de software overdraagbaar van de ene computer naar de andere. Besturingssystemen als UNIX zijn min of meer overdraagbaar. MS-DOS is slecht overdraagbaar en werkt zonder veel kunst- en vliegwerk alleen op IBM compatible PC's! Een aantal andere operatingsystemen kan echter wel een MS- DOS-omgeving emuleren (= nabootsen).

De **Shell** vormt een soort gebruiksvriendelijke schaal tussen de man/vrouw achter de computer en de moeilijke taal van het besturingssysteem. In het eenvoudigste geval is er sprake van een zoge-

Compac

A 231 035 - 260633

De nieuwe Seikosha 24 naalds printer SL-230 wordt door Compac getoond. De printer is voorzien van een nieuw ontwikkelde printkop met een parallelle naald opstelling en heeft een printsnelheid van 277 tekens in draft en 92 tekens in letterkwaliteit over een breedte van maximaal 136 karakters bij 10 cpi.



De printer heeft een programmeerbare insteekkaart, waarop de gebruiker zijn persoonlijke printer instellingen kan opgeven.

De Seikosha SL-230 is standaard voorzien van 9 verschillende schriftsoorten, heeft zowel een parallelle als een seriële ingang en is uitgerust met een IBM Proprinter XL-24 en een Epson LQ-1050 emulatie. De input buffer meet 64 KB en kan eventueel uitgebreid worden met 64 KB.

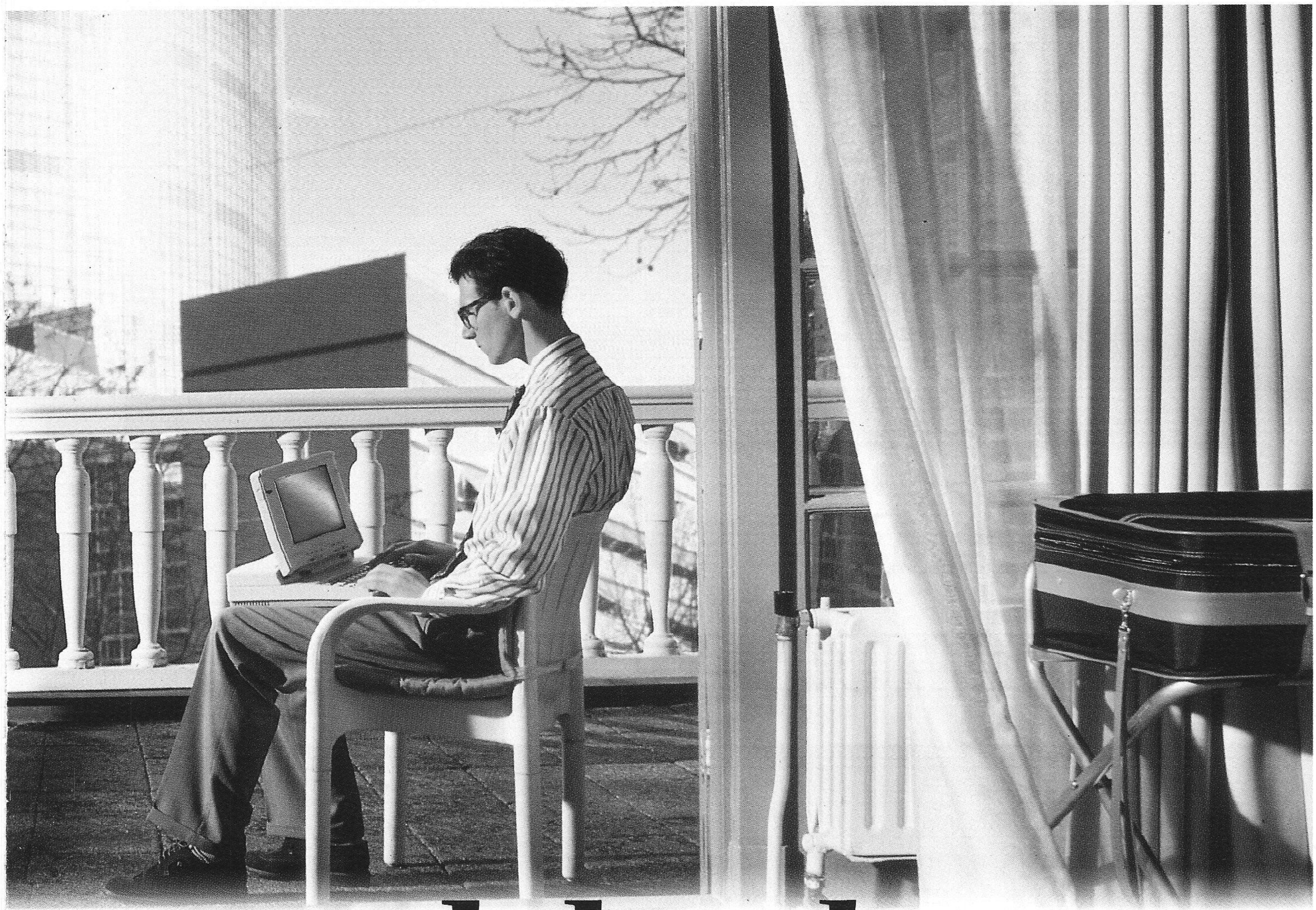
Compucon

D 720 03465 - 70484

Compucon richt zich sterk op het nieuwe besturingssysteem Vmos/3, een multi-tasking besturingssysteem voor de 386 processor. Vmos/3 is een produkt van StarPath Systems uit Michigan (VS) en is in principe volledig DOS compatible. Door het gebruik van virtueel geheugen (geheugeninhoud komt op de harde schijf terecht zonder dat de software er wat van merkt) kunnen meerdere programma's gelijktijdig actief zijn.

Daarnaast toont Compucon, als leverancier van de PC's van Advanced Logic Research, de nieuwste modellen van deze leverancier, onder andere de Flex-Cache 386 33 MHz machine.

Special



nu ook leesbaar.

Een betrouwbare micro die ik overal mee naar toe kan nemen, lijkt me heel handig. Stuur mij documentatie.

Naam: ________________________________ m/v

Bedrijf: ________________________________

Adres: ________________________________

Postcode: ______________ Plaats: ______________

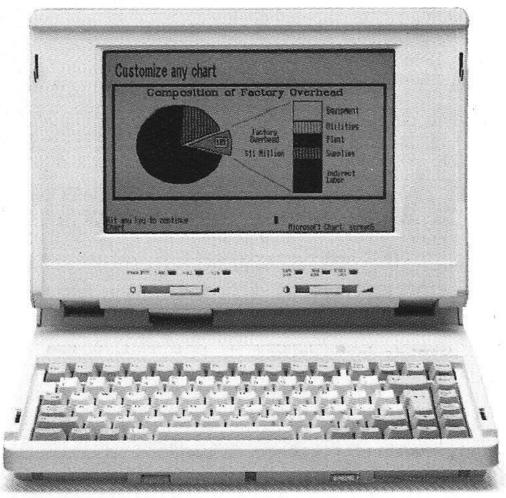
Telefoon: ________________________________

Huidig computermerk: ________________________________

Belangrijkste toepassingen: ________________________________

Stuur de bon in een envelop zonder postzegel naar: Tulip Computers Nederland B.V., Antwoordnummer 13039, 5200 VE 's-Hertogenbosch.

Tulip Computers, Hambakenwetering 2, 5231 DC 's-Hertogenbosch. Telefoon 073-405333, telex 50316 tulip nl, fax 073-421915. PBI 1



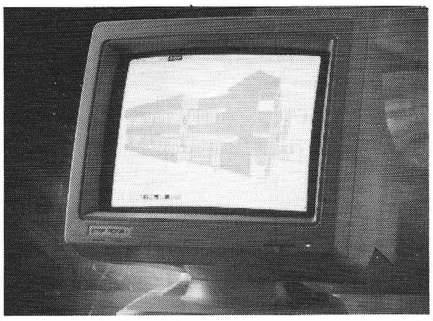
Tulip® computers

De naam voor Europese kwaliteit.

Etap Information Techn.

B 513 *076 - 417800*

Etap laat een aantal High Resolution monitors zien. Niet alleen ondersteunen de A4 en A3 beeldschermen en bijbehorende videokaarten de meeste softwarepakketten op het volledige scherm, ook is het mogelijk dat de gebruiker het beeldscherm indeelt zoals hij/zij dat zelf verkiest.



Met WordPerfect 5.0 bijvoorbeeld is het mogelijk een grafische indruk van de pagina te krijgen op het A4 scherm. Bij een A3 scherm staat de pagina op de ene helft, terwijl op de andere helft de tekst geredigeerd kan worden. Ook bij MS-Word 5.0 is het mogelijk previews te krijgen, exact zoals het afgedrukt wordt.
Voor AutoCAD 10.0 levert Etap een speciale driver voor de Pictor kleurenbeeldschermen.
De beeldschermen hebben een hoge beeldfrequentie (75 Hz non-interlaced) en gebruiken hoog-luminante witte fosfor en positieve video. Er kunnen 51, 66 of 72 regels tekst weergegeven worden. Hoe kleiner het aantal regels, hoe groter de letters en dus hoe hoger het leescomfort.

Arcade O. Harris

W 450 *03480 - 17590*

Deze specialist op CAD gebied demonstreert op de beurs de nieuwste versie 1.4 van het Arkey/Arcos 2D/3D CAD pakket. Ook wordt voor dit pakket een facility management faciliteit gepresenteerd.
Arkey/Arcos is een op de bouwsector gericht ontwerp- en produktiepakket en wordt gebruikt door zowel architecten, meubel- en sanitair ontwerpers als installatiebedrijven.
De laatste versie

naamd **Command Line-interface**. De naam zegt het al; de gebruiker/ster typt achter een bepaald teken (prompt in de vorm van een **>**, **.**, **$**, **%** e.d.) een opdracht op de commandoregel in. Na het geven van een Enter/Return wordt deze opdracht uitgevoerd. Bijvoorbeeld bij MS-DOS: COPY A:*.* C: kopieert de inhoud van diskette A naar de harde schijf C. Andere typen interfaces zijn:

- Het **menu**. Er verschijnt een reeks keuzemogelijkheden op het scherm die via de cursortoetsen, met de muis of via het intypen van een letter geactiveerd kunnen worden.
- Een **WIMP omgeving**. Deze gebruikersomgeving bestaat uit vensters (windows), Ikonen (symbolen voor opdrachten en files), Muisbesturing en trekmenu's (pull-down menu's). Met de muis kan de gebruiker alle gewenste opdrachten geven, de schijfinhoud en draaiende programma's bekijken en allerlei handige schermphulpjes activeren. Bekende schermhulpjes zijn oa. een rekenmachine (calculator), notitieblokje (notepad) en agenda. Bekende WIMP omgevingen zijn bijvoorbeeld, MS-Windows voor MS-DOS, de Workbench voor de Amiga en OSF Motiv voor UNIX.

De **Utilities** zijn miniprogramma's die standaardopdrachten uitvoeren. Bijvoorbeeld files kopiëren, schijven formatteren, het beeldscherm afdrukken, backups maken, een inhoudsopgave op de monitor zetten, bestanden editen, het uitvoeren van diverse netwerk- en beveiligingstaken. Hoe recenter de versie van het besturingssysteem hoe meer en geavanceerder zijn de meegeleverde utilities.

De gebruiker kan ook **zelf OS programma's schrijven**. Daar de boven op het Kernel liggende OS-lagen gebruik maken van een of andere programmeertaal, kunt u zelf opdrachten aan elkaar koppelen. Mogelijk zijn oa. batches, macro's, utilities, scripts en complete toepassingsprogramma's.

De communicatie tussen de hogere lagen en het Kernel verloopt via de **system calls** en de daaropvolgende **returns**. De draaiende software roept via het OS een bepaalde Kernel functie aan en na uitvoering of verhindering daarvan geeft het Kernel een antwoord over de stand van zaken.

De hier geschetste werking van een OS is uiteraard zeer globaal en kan per besturingssysteem sterk verschillen. Het indelingsprincipe geldt echter wel universeel.

Files en directories

Bestanden of **files** zijn de vorm waarin de software en de bijbehorende gegevens (data) worden opgeslagen. Elk besturingssysteem gebruikt daarvoor een bepaald eigen **format**. Dit format bepaalt hoe de data weggeschreven en weer ingeschreven kunnen worden. Bij uitwisseling van gegevens tussen twee besturingssystemen kunnen problemen ontstaan doordat het ene de files van het andere systeem of zelfs de gehele diskette niet kan lezen.

Op een schijf gaan al gauw vele honderden files. Om het overzicht te kunnen behouden worden die bestanden in hoofdstukken verdeeld. Men noemt de grote hoofdstukken **directories** en hun paragrafen de **subdirectories**. De meeste systemen gebruiken een boomstructuur. Bovenaan die boomstructuur staat de **root-directory** met de essentiële gegevens voor het opstarten en wie waar op het systeem te keer mag gaan. Daaronder kommen de verschillende directories en subdirectories. Voor het lezen van een bepaald bestand zal het OS eerst de desbetreffende directory en/of subdirectory moeten openen. Anders kan de computer er niet bij.

MS-DOS

Het meest bekende OS in de PC-wereld is **MicroSoft DOS** (MS-DOS), in de IBM-versie **PC-DOS**. Doordat er zoveel software voor MS-DOS bestaat en de daarvoor benodigde hardware relatief goedkoop is, heeft het gebruik een ongekende vlucht genomen. Achteraf bezien is dit DOS een tamelijk achterlijk OS. De beperkingen zitten hem ondermeer in:

- Het beperkte **RAM gebruik**. Zonder extra drivers kunnen de oudere versies met niet meer dan 640 KB uit de voeten. Andere OS kunnen vele MegaBytes aan vrij RAM hanteren
- De oudere DOS-versies kunnen zonder diskmanagers met geen grotere harddisks dan **32 MB** werken
- Op **netwerken** en **multi-tasking** werd niet gerekend, terwijl de nieuwe 386/486 PC-generatie hiervoor nu juist krachtige mogelijkheden biedt
- De **programmeermogelijkheden** zijn relatief beperkt
- MS- en PC-DOS zijn primair voor de verouderde **8086/8088 CPU** geschreven

Natuurlijk is achteraf klagen altijd gemakkelijk. Andere gebruiksvoorwerpen zoals auto's hebben ook een complete evolutie doorgemaakt. En wie rijdt er nu nog in een DOS 1.X-versie T-Ford? Bovendien bedienen de huidige 3.X en 4.X- versies de gewone stand alone PC-gebruiker naar volle tevredenheid. En de lage gebruikskosten maken immers veel goed. Alleen de power-, multi-tasking- en netwerk-user ontdekt al gauw de beperkingen.

De ontwikkelingen staan ook bij MS-DOS natuurlijk niet stil. Men wil niet meer wachten totdat het document op de printer is uitgedraaid. Verwijs de printercontrole naar de achtergrond zodat je in de tussentijd met iets anders door kunt werken. Kortom, **multi-tasking**, waarbij meerdere programma's tegelijk op één machine kunnen draaien.

Op een 80386 PC is multi-tasking goed te doen. Met behulp van een listige truc wordt die ene Intel 80386 of 80486 in tal van virtuele 8086 microprocessoren opgedeeld. Elk draaiend programma krijgt een eigen werkomgeving (RAM plus virtuele 8086) toegewezen en denkt dat hij het koninkrijk alleen heeft. In de praktijk dient de software echter wel op zijn CPU-beurt te wachten. Dankzij de snelheid van de 386/486 CPU met hardware en de goede timing van de multi-tasking software zijn de wachttijden bij twee tot zes programma's gering en lopen de systeemprestaties niet al te veel terug. Voorbeelden van multi-tasking software zijn oa. Windows 386, DOS 386, VM 386, DESQ 386 en PC-MOS.

Netwerken en **multi-user-gebruik** leven bij de gratie van additionele software. Dankzij de komst van zware PC's (snelle CPU's, groter geheugen en harddisk capaciteit) en LAN-kaarten met bijbehorende software, kunnen nu redelijk snelle PC-netwerken gecreëerd worden. De software, bijvoorbeeld Novell of 3COM, voorziet in de benodigde extra drivers.

OS/2

Het oorspronkelijk voor de IBM PS/2 lijn en compatible AT's bedoelde **Operating System/2** (OS/2) mikt op de power-user op de stand-alone PC. De sterke punten zijn oa.:

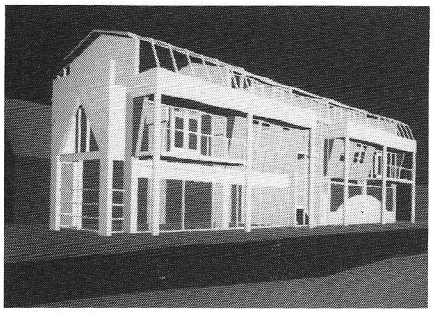
- **Multi-tasking**
- **Snelle interproces communicatie**
- Een **geïntegreerde WIMP omgeving** in de vorm van de Presentation Manager
- Een **vrij adresseerbaar RAM van 16 MB**
- Geen **32 MB harddisk begrenzing meer**
- Meer geënt op de **modernere CPU** (80286 en 80386)

Aanvankelijk deed OS/2 weinig. De opeenvolgende OS-versies kwamen te laat uit, waren incompleet en duur. OS/2 compatible programmatuur was nauwelijks te krijgen en de benodigde extra hardware uitbreidingen (4 MB RAM) waren duur. Inmiddels zijn de prijzen gedaald, de OS/2-versies compleet, kosten de RAM chips minder en bieden de nieuwe OS/2 versies van Paradox, R-Base, Q&A en andere DOS virtuozen nu ongekende mogelijkheden op de stand-alone PC. Zij die primair een multi-user systeem zoeken, kunnen echter beter naar een ander OS uitkijken.

UNIX / XENIX

Vaak wordt de indruk gewekt dat **UNIX** iets geheel nieuws is of een

1.4 heeft een aantal nieuwe teken- en ontwerpfuncties erbij gekregen. De nieuwe facility management mogelijkheid, het beheer en onderhoud van gebouwen, gekoppeld



aan CAD biedt de mogelijkheid om met CAD opgebouwde tekeningen te gebruiken als basis voor beheer en onderhoud.

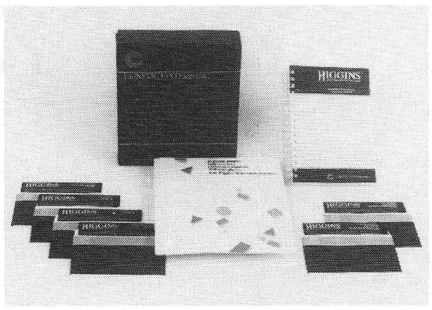
Infotheek Systems

E 312 ***071 - 256356***

Behalve ontbijtseminars zal op de stand van Infotheek een drietal thema's centraal staan: netwerken en datacommunicatie, desktop publishing en desktop presentations, computerdienstverlening.

De bezoeker kan kennismaken met de mogelijkheden en werking van verschillende netwerkconfiguraties.

Op communicatiegebied is het nieuwtje de J&L Chatterbox. Bij steeds meer bedrijven wordt het gangbaar dat werknemers, klanten e.d. in kunnen bellen op de bedrijfscomputer. Om inbellers niet



te frusteren door bezette lijnen dient de Chatterbox. Hiermee kunnen maximaal 32 inbelstations in een behuizing op te nemen.

'Higgins' van Conetic Systems is netwerksoftware die is opgezet rondom een relationele database en die een grote hoeveelheid gegevens overzichtelijk beschikbaar houdt voor een brede groep gebruikers. Higgins

Special

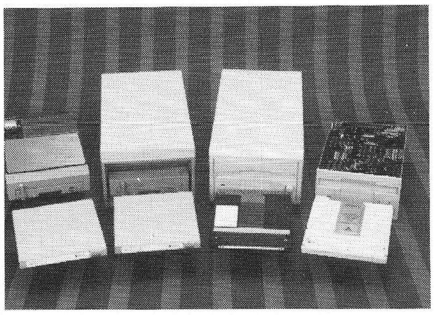
kent diverse modules, oa voor het bijhouden van agenda's, klantenbestanden, archiveren, elektronische post, etc.

Intra Electronics

D 707 040 - 836455

Intra Electronics levert vooral produkten op het gebied van de computer periferie, subsystemen en elektronische componenten. Op de beurs presenteert het bedrijf oa een Rewritable optische disk drive van Ricoh, met een opslagcapaciteit van 600 MB. De 5.25" drive is uitgerust met een SCSI interface en is compatible met de Ricoh WORM.

Nieuwe harddisk drives zijn er van verschillende fabrikanten. De nieuwe Miniscribe 9780 harddisk drive heeft een capaciteit van maar liefst 676 MB. Deze drive heeft de 'volle hoogte' en beschikt over 8 schijven en 15 koppen. Intra geeft



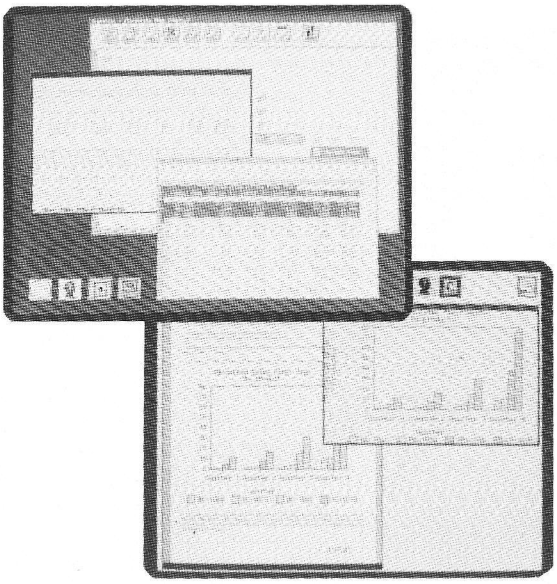
voor deze drive op, dat de gemiddelde tijd tot het optreden van een fout (MTBF) 50.000 uur is. De 9780 kan zowel volgens de ESDI als de SCSI techniek werken, met een data-overdrachtssnelheid van 15 Mbit/sec. De Miniscribe lijn loop van 20 MB tot 676 MB modellen.

Van C._Itoh zijn er de 3,5" harddisk drives. Deze zijn vooral bedoeld voor laptops en portables. Floppy disk drives komen van Y-E Data.

Van het Duitse merk Infosys zijn er geheugen uitbreidingskaarten, batterijgestuurde RAM-disk kaarten, videokaarten etc. Ook laat Intra Electronics een serie combinatiekaarten zien, bijvoorbeeld een kaart met geheugen uitbreiding plus parallelle en seriële poort en EGA.

nieuwe standaard is geworden. De waarheid is echter dat het om een 'oude rakker' uit de zeventiger jaren gaat, die men de laatste tijd probeert op te krikken en tot universele standaard te verheffen.

In tegenstelling tot MS- en PC-DOS is deze old timer qua specificaties nog altijd indrukwekkend. Multi-tasking, multi-user, uitgebreide communicatie mogelijkheden, interne security, vele honderden standaard utilities, uitgekiend onbeperkt RAM gebruik, besturing van zeer grote harddisks, interproces communications, tegenwoordig ook een WIMP shell, programmeren in de taal C, de mogelijkheid tot MS-DOS emulatie in afzonderlijk draaiende processen, een prima filestructuur en een groot aantal tools (software gereedschappen) bieden vrijwel alles wat de gebruiker begeert. Doe daarbij nog een schat aan software, gebruikerservaring, wereldwijd gebruik en een hoge mate van overdraagbaarheid van de programmatuur en er ontstaat een zeer krachtig en veelzijdig OS.



Met DOS Windows biedt het Sun 386i workstation een brug tussen Unix en DOS. Meerdere DOS programma's kunnen tegelijkertijd in verschillende vensters draaien.

Toch gebruikt nog niet iedereen UNIX. Dat vindt zijn oorsprong in de volgende redenen:

- UNIX is domweg **te duur** voor de kleine gebruiker. Wie het allemaal niet nodig heeft betaalt zich ten onrechte blauw aan terminals, fileservers, het OS zelf en de software.
- Is **nog niet zo'n echte standaard** als het lijkt. Er komt steeds meer uniformiteit. Toch blijven er wel degelijk compatibiliteitsverschillen.
- **Sommige software is (nog) niet onder UNIX verkrijgbaar**
- Een aantal bedrijven heeft al **te veel in MS-DOS geïnvesteerd**.

De verwachtingen zijn dat UNIX een flink stuk koek van de zakelijk markt voor zich zal gaan opeisen. Een belangrijke implementatie lijk UNIX op RISC machines te worden. Enkele leveranciers zijn hier al tamelijk ver mee. Of UNIX ook voor de kleinere gebruiker interessant gaat worden zal de tijd ons leren.

Een computer zonder OS is als een ijscoman zonder ijskarretje. Er gebeurt, behalve het roepen om het karretje, niets. Voeg een besturingssysteem toe en de gebruiker kan de computer opdrachten geven, toepassingssoftware laten draaien, met andere computers communiceren en de randapparatuur bedienen. Welk operating systeem precies bij u past hangt van de hardware en het toekomstig gebruik af. MS/PC-DOS voldoen al of niet met uitbreidingen nog altijd prima voor de kleine gebruiker. De power-user en multi-tasker kan eens naar OS/2 kijken hoewel deze laatste ook met de genoemde DOS-extensies uit de voeten kan. Voor de grotere zakelijke markt is UNIX duidelijk in opkomst en zal wellicht een nieuwe serieuze software standaard worden.

U.S

Het computergeheugen wordt verdeeld in een intern en extern geheugen. Kenmerkend voor het interne geheugen is de data opslag in chips. Dit in tegenstelling tot het externe geheugen, dat gebruik maakt van magnetische of optische informatiedragers op schijf of tape.

Drives

Het externe geheugen van uw computer



De Tandon 386/33 heeft een geïntegreerde 'Personal Data Pac' verwijderbare harddisk

Bij het opstarten van de PC bevat de computer slechts een minimale hoeveelheid informatie. In het **Read Only Memory** (ROM - permanent in geheugenchips aanwezige informatie) staat slechts hoe de machine een aantal basistesten moet uitvoeren en waar de PC naar de opstartroutines en het operatingsysteem moet zoeken. Is de PC klaar met checken, dan wordt via een bootroutine informatie vanaf de diskette of harddisk naar het **Random Access Memory** geladen (RAM - tijdelijk in chipbanken opgeslagen informatie die verdwijnt na wissen of het uitschakelen van de stroom). Deze informatie bestaat uit het DOS (of ander besturingssysteem), het gebruikersinterface of een menu. Slechts bij een klein aantal PC's, zoals de Amiga en Atari, is het besturingssysteem reeds in het ROM ingebakken en behoeft dat bij de start niet van de schijf ingeladen te worden.

Het RAM bevat **informatie in vluchtige staat**. Onder dat 'vluchtig' dient u 'tijdelijk' te verstaan. De RAM-data worden voortdurend aangepast, ververst en van/naar de schijven (diskettes, harddisks) geladen. Wordt de stroom uitgeschakeld dan is het RAM leeg en gaat alle daarin opgeslagen informatie verloren. Een bekende beginnersfout die al menige krachtterm heeft ontlokt. Zeg niet dat het u nooit is/zal overkomen.

De enige manier om de programmatuur en gegevens in het RAM te bewaren is ze naar een magnetische of optische informatiedrager weg te schrijven. Elke PC wordt standaard met minimaal één **diskdrive** geleverd. In die diskdrives passen magnetische schijfjes, de diskettes, waarop de programma's en bijbehorende data staan. Dankzij deze diskdrives is het mogelijk om software van de ene PC naar de andere over te bregen en programma's en data los van de computer te bewaren. Behalve de diskdrives omvat het drive repertoire voor de PC's nog de harddisks, tapestreamers en optische drives. We zullen ze stuk voor stuk onder de loupe nemen.

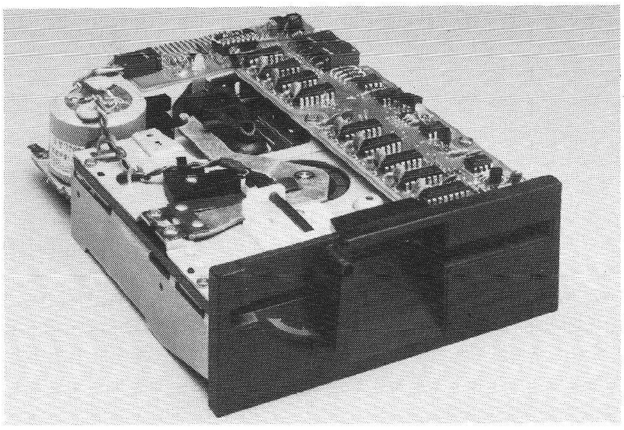
Diskdrives

Kenmerkend voor de **diskdrive** is dat er uitneembare diskettes, 'floppies', in zitten. Dat floppy slaat op de flexibele aard van het magneetschijfje dat in een plastic hoesje, de jacket, zit. Het hoesje dient om de schijf tegen stof, vuil en fysieke schade te beschermen. In het midden zit een gat voor de aandrijving door de drive en aan de voorzijde een al of niet (met een weg te schuiven afdekplaatje) bedekte lees/schrijfsleuf.

De drivemotor zorgt er voor dat de diskette een x-aantal honderd toeren per minuut maakt. Standaard is ongeveer 300 per minuut, maar er zijn diskdrives die het wat langzamer of juist veel sneller doen. Dankzij die hoge rotatiesnelheid kunnen de over de diskette heenscherende

lees/schrijfkoppen in korte tijd een groot aantal datasporen, **tracks**, en hun sectoren bewerken. De term **leesschrijfkop** geeft al aan dat de koppen zowel data van de diskette naar het RAM kunnen lezen als van het RAM naar de schijf kunnen wegschrijven.

Alvorens een diskette bruikbaar is dient deze **geformatteerd** te worden. Formatteren is niets anders dan een magnetische indeling en boekhouding (index) maken. De diskette wordt verdeeld in een x-aantal tracks en een index-spoor (track 0) dat bijhoudt waar de weggeschreven software en data te vinden zijn. Het besturingssysteem beweegt de koppenarm aan de hand van de indexgegevens naar de juiste lees/schrijfplaats.



Shugart's SA200 5.25-inch Minifloppy™ disk drive

Om een diskdrive in de computer te installeren dient u met de volgende punten rekening te houden:

- De **besturing en communicatie** met de drive verloopt via een **controllerkaart**. Deze controller verzorgt het stoppen/starten van de drive, de stuursignalen en de in/output van data. Uit de controllerkaart komt een lintkabel met één of meerdere connectoren voor aansluiting op de bus aan de achterzijde van de drive. Het behoeft geen betoog dat controller en drive met elkaar compatibel dienen te zijn. Let hier op bij het gebruik van verschillende typen diskdrives door elkaar! Een PC XT kan bijvoorbeeld zonder upgrade geen AT diskdrives aansturen.
- Het **besturingssysteem** (OS) dient met het drivetype te kunnen werken. Met name bij de oudere DOS-typen wil het werken met de nieuwere 3.5 inch drives wel eens problemen geven.
- Gebruik het **juiste type I/O lintkabel**
- Is er **ruimte in het drive chassis** voor de nieuwe diskdrives? Zo ja, beschikt u over de goede schroefjes, geleiderails en voorfrontjes?
- Is er nog een **voedingskabeltje** voor de diskdrive vrij?
- Last but not least dient uw PC **'te weten'** dat er een extra drive in de machine zit. Was er bijvoorbeeld eerst alleen sprake van een diskettestation **A** dan zit er nu ook een **B** in. Dit zal aan het **setup programma** of via dipswitches (sommige oude XT's) gemeld moeten worden.

Jammer genoeg is er weinig uniformiteit bij de PC diskdrives. Er wordt onderscheid gemaakt bij:

- De **fysieke afmetingen** van de drive. Gangbaar zijn de maten 3.5 en 5.25 inch. Een diskettestation werkt alleen met de schijven die daar in passen. Bij de montage in de computer zult u met de maten van de diskdrives en de ruimtelijke afmetingen van het drive chassis rekening dienen te houden.
- Het **dataformat**. De klassieke **5.25 inch PC diskdrive** kon 360 KiloByte (KB) aan data bevatten. Enkele jaren later kwam daar nog een 720 KB versie bij die het niet echt gemaakt heeft. Wel ingang vond het **1.2 MB AT-model** die gezien de grote opslagcapaciteit ideaal is voor grotere programma's, datafiles en het backuppen van bestanden. Nieuwer en snel in opmars zijn de kleinere **3.5 inch drives** in de formats 720/800 KB en 1.44 MB. De bijbehorende diskettes zitten in een hardplastic jasje en de lees/schrijfsleuf wordt door een, via een magneet weg te schuiven, afdekplaatje gesloten. Trek dit type diskettes nooit uit een draaiende drive om

Laser Computer Europe

A 241 ***071 - 410801***

Laser Europe siert zijn stand met een serie nieuwe computers, van XT's tot 386 Towers.
De XT/SL is voorzien van een 10 MHz 8086-1 processor, 640 KB RAM, een seriële en een parallelle poort, vier slots en twee 3,5" fixed disk units. Deze XT kan oa gebruikt worden als diskloos station voor netwerktoepassingen.
De Laser AT/2 en de AT/3 zijn 80286 computers met een resp. 12 en 16 MHz processors. Het RAM is standaard 640 KB en is op het moederbord uit te breiden tot 3072 KB en met kaarten tot 16 MB. In de AT/2 zit een fixed disk unit en een 20 MB harddisk en in de AT/3 een 40 MB disk. De AT's worden geleverd met een 14" flatscreen monitor, MD-DOS 4.01, GW-Basic, PC Tools 5.* en een printerkabel. Prijzen variëren van *f* 3899,- tot *f* 4650,- ex. BTW.
De 386 serie bestaat uit de Laser 386 SX, de 386 Desktop en de 386 Tower. De SX is voorzien van de 386 32-16 bits 16 MHz processor, 1 MB RAM dat op het moederbord is uit te breiden tot 8 MB, zeven AT slots, een diskdrive en een 40 MB harddisk.
De Desktop en Tower modellen van de 386 hebben een 25 MHz processor, zero waitstate en met een cache van 64 KB. Het 2 MB geheugen is uit te breiden tot 10 MB. De hard disks hebben een capaciteit van 40 tot 65 MB. Met monitor, keyboard en diverse software bedragen de prijzen van *f* 8.999,- voor de Desktop tot 9.899,- voor de Tower, ex. BTW.

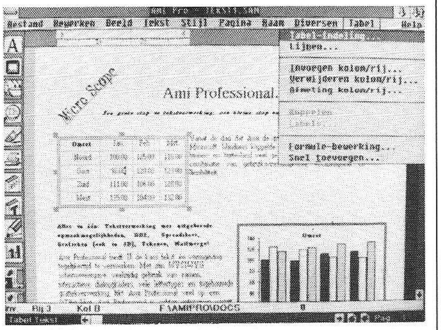
Micro Scope

W 460 ***010 - 4563799***

Micro Scope lanceert op de beurs 'Amí Professional' voor de grafische gebruiksomgeving van MS-Windows. Het is de high-end versie van het Amí tekstverwerkingspakket van Samna Corp. De professionele versie biedt veel extra functies, waaronder een thesaurus, tekenprogramma, mail-merge, voetnoten, macro's en tabelverwerking met spreadsheet functies. De tabelgegevens kunnen in grafieken worden omgezet. Via een in- en uitvoer functie

Special

kunnen andere aan MS-Windows gekoppelde applicaties in Amí Professional worden gewijzigd. De Engelstalige versie verschijnt in oktober, de Nederlandstalige in november.



Andere Windows applicaties die Micro Scope toont zijn Amí (de gewone versie), ComfoWare, The SQL System en Clip & Connect.

Olivetti Nederland

D 714 ***071 - 319931***

Olivetti toont een interessante ontwikkeling afkomstig van AT&T, met de naam Premises Distribution System. Dit PDS is de infrastructuur voor alle typen communicatie in gebouwen of gebouwencomplexen. PDS kan zowel analoge als digitale spraak, dataoverdracht (300 bps tot 200 Mbps), beelden van fax, terminals of plotters, videobeelden en sensorsignalen verwerken. PDS is een totaalsysteem, dat onafhankelijk van protocollen is en slechts een klein paneel nodig heeft voor de administratie van het systeem. De bekabeling is gestandaardiseerd evenals de aansluitingen. Daardoor is het verhuizen van apparatuur erg makkelijk en hoeft er alleen maar een andere verbinding gelegd te worden op het administratiepaneel.

Ormas

Z 466 ***03403 - 90911***

Affinity is een geïntegreerd pakket voor kantoorautomatisering dat zowel onder DOS als Unix draait. Het is een modulair systeem dat via een Ethernet koppeling kan zorgen voor multi-user mogelijkheden over verschillende typen computers. Ormas laat ook een nieuw pakket zien voor de groothandel, dat, waarschijnlijk met verwijzing

schade aan de schijf en drive te voorkomen.

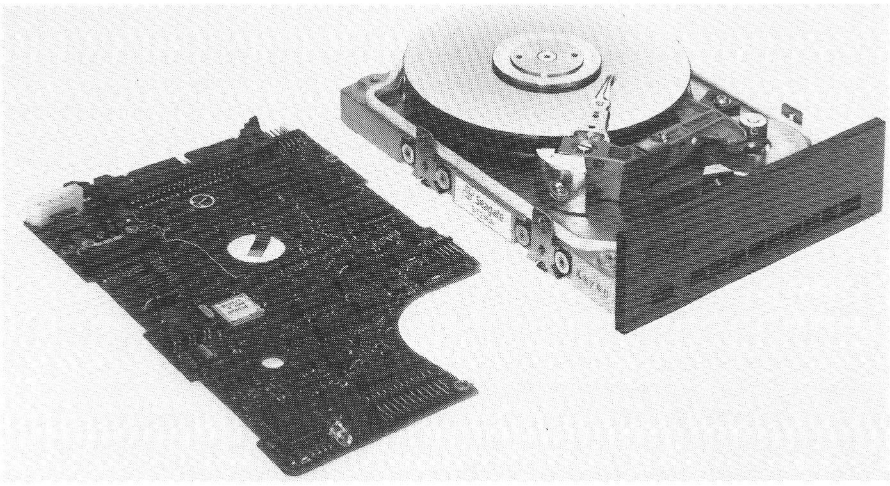
Behalve de hier genoemde diskdrivetypen komen er langzaam steeds meer high speed en high capacity drives bij. Deze diskdrives met bijbehorende speciale floppies gaan de data opslagcapaciteit van een kleine harddisk evenaren. Een probleem is hun speciale format, dat alleen door dat ene specifieke drivetype gelezen kan worden.

Diskettes plaatst u altijd met de lees/schrijfsleuf naar voren en het label naar boven in de drive. Bij de 5.25 inch drives dient de gebruiker de diskette met het handeltje te ver-/ontgrendelen. Een 3.5 inch model vergrendelt zich vanzelf. Het ongrendelen gebeurt door bij **stilstaande drive** op het kleine ejectknopje te drukken.

Harddisks

Tegenwoordig worden steeds meer PC's met een vaste schijf of **harddisk** verkocht. Dat 'hard' slaat op het feit dat deze drives magnetisch gecoate metalen platen in plaats van flexibele plastic folies gebruiken. Een harde metalen plaat kan veel sneller draaien dan een flexibele. De lees/schrijfkop kan dus per tijdseenheid meer data inlezen of wegschrijven dan bij een diskdrive. **Snelheid** is één van de belangrijkste redenen om voor een harddisk te kiezen.

De tweede belangrijke reden is de **zeer grote data opslagcapaciteit**. Voor circa *f* 1000,- koopt u al een 40 MB model en daar gaan zo'n 112 diskettes van 360 KB op. Een dergelijk grote opslagcapaciteit maakt het mogelijk om software en de bijbehorende databestanden op één en hetzelfde medium te gebruiken. Dat spaart veel tijd en lastig gewissel met diskettes.



ST296N van Seagate biedt een opslagruimte van 84 MB. De toegangstijd bedraagt 28 millisec.

Ook bij de aansluiting van een harddisk dient u weer rekening te houden met de bij de diskdrives genoemde installatie criteria. Daar komen echter nog drie punten bij:

- De harddisk moet **qua snelheid** bij uw PC passen. Met het steeds sneller worden van de microprocessoren kan een harde schijf met te trage lees/schrijftijden het gehele systeem verlammen. Kies daarom bij een AT of 386/486 machine voor een drive met een accestijd van minimaal 40 millisec. en liever nog 28 ms of minder.
- Het **gebruikte besturingssysteem**. Oudere MS-DOS versies (t/m 3.3) hebben moeite met een opslagcapaciteit van meer dan 32 MB. Dan zijn er extra drivers nodig om toch de volle capaciteit te kunnen benutten of een 40 MB drive C gewoon in een 20 MB C en 20 MB D op te delen. UNIX/XENIX, OS/2 , DOS 4.X en een aantal andere besturingssystemen kunnen rustig vele honderden MB's aanspreken.
- Indien de leverancier de klus niet voor u opknapt zult u de **harde schijf** zelf moeten **formatteren**. Dit proces staat nauwkeurig beschreven in het DOS manual en hopelijk ook in de bij de disk gele-

verde gebruiksaanwijzing. Bij nauwkeurig werken kan iedereen dit zelf doen, zij het dat het bij 40 MB en meer wel een uurtje kan duren.

Tegenwoordig komen er steeds meer harddisks die niet vast in de PC geïnstalleerd hoeven worden, maar uitneembaar zijn. Er zijn ruwweg twee verschillende typen:

- **In zijn geheel uitneembare harddisks**. Er zijn harddisks die in een speciaal drive chassis gaan en een aantal leveranciers brengt sledes voor alle gangbare drivetypen uit. Meestal is vergrendeling van de disk met een sleuteltje of codewoord mogelijk.
- Harddisks waarbij alleen de **schijven zelf** worden uitgenomen.

Uit het oogpunt van beveiliging is de uitneembare harddisk een belangrijke ontwikkeling. De gebruiker kan nu de harde schijf uitnemen en in de safe opbergen. Verder zijn er nog de **draagbare harddisks**. Meestal gaat het om **'bomvrije'** types die een ruw zakenreisje weten te overleven.

Als laatste harddisktype de **hardcard**. Bij een hardcard zitten de controller en harddiskdrive op dezelfde kaart. Steek de kaart in een vrij PC-slot, formatteer, draai de setup en de harddisk werkt. Geen geknoei met kabeltjes meer en het bespaart driveruimte. Een nadeel is wel dat hardcards doorgaans iets trager zijn dan de conventionele losse harddisks met controller.

Dankzij de sterk gedaalde prijzen, toenemende snelheid en opslagcapaciteit in MegaBytes is de harddisk ook bij de kleine gebruiker populair geworden. Voor de PC, Amiga en Atari kost een goed 20 MB model minder dan *f* 1.000,-. Pas op met bodemprijs PC-typen, want die zijn meestal alleen voor de XT bedoeld.

Tapestreamers

De tapedrive of **tapestreamer** dankt een groot deel van zijn populariteit aan de niet-uitneembare harddisk. Backuppen naar floppies is een tijdrovend en diskjockey-achtig karwei dat voortdurend toezicht nodig heeft. Met een tapedrive voor uitneembare datacartridges gaat dat allemaal veel comfortabeler en sneller.

Een tapestreamer is in feite niets anders dan een high speed tapedrive. Een controllerkaart in de PC en de meegeleverde backup software zorgen voor een perfecte overschrijving van de harddisk data naar de tapecartridge. Met deze datacartridge in de safe bent u gedekt tegen ongevallen en gebruikersfouten op de harde schijf.

Behalve een echte tapestreamer is het ook mogelijk om via een speciale controllerkaart de videorecorder als backup- medium te gebruiken. Wellicht trager, maar ook aanzienlijk goedkoper wat de cartridges betreft.

Tapestreamers zijn de laatste tijd flink in prijs gedaald. Voor rond de *f* 1.000,- koop je tegenwoordig al een 40 MB inbouwmodel. Sommige PC's worden zelfs al standaard met tapestreamer geleverd.

Behalve tapestreamers zijn er ook nog zogenaamde **tapedrives** te koop. Tapedrives worden primair als data- en niet als backupdrive gebruikt. Qua prestaties scoren deze drives tussen de diskdrives en de harddisk.

Optische drives

De toekomst is volgens insiders aan de **laserdisk**. Een glimmend, door een laserstraal uit te lezen CD-schijfje kan een complete encyclopedie aan informatie bevatten. De opslagcapaciteit varieert van 650 MB tot vele GigaBytes. Behalve gewone programmatuur en datafiles passen op deze computer-CD's ook gedetailleerde grafische beelden, foto's, animaties, life video, muziek, gesproken commentaar, geluidseffecten en interactieve lesprogrammatuur.

Voor het afspelen van deze **CD-ROM**, **CDI**, **CDV** en wat men nog meer aan optische disks bedacht heeft, zijn een speciale drive en controller nodig. De kosten daarvan zijn vooralsnog hoog. Andere tijdelijke bezwaren zijn:

- De **relatieve traagheid** van de drives

naar 1992, Eurohandel heet.
Van Sharp zijn er diverse nieuwe produkten te zien, waarvan wij slechts de drie draagbare computers (2 XT's en een AT) en de 'persoonlijke databank' IQ-7000 willen noemen. Dit laatste 'rekenmachientje met computercapaciteit'



kan via een seriële verbinding naadloos samenwerken met een PC. Tenslotte van Sharp een kleurenscanner voor koppeling aan de PC. De X-300 haalt een resolutie van 300 dpi en kan 260.000 kleurnuances in 256 gradaties verwerken.
Het merk MDS levert via Ormas ook een noviteit: de MDS 3065 laserprinter geschikt voor een aantal grotere computers zoals IBM mainframes.

PC Choice

A 244a ***020 - 6626288***

PC Choice heeft aan zijn produktenreeks een non-dedicated printserver utility toegevoegd. Met Print+ voor 3Com-LAN's wordt het mogelijk om printers op lokale werkstations aan te sluiten, zodanig dat elke gebruiker deze printer kan gebruiken. De utility ondersteunt 3+ en 3+Open en is daarin volledig geïntegreerd. Ook ondersteunt het alle DOS, BIOS en 3Com MS/REDIR commando's.

Positronika

A 241a ***040 - 416355***

Deze grote distributeur heeft een hele reeks nieuwe produkten, sommige al eerder aangekondigd, andere echt nieuw (zoals het wel vaker gaat met noviteiten). Wat betreft drives e.d. zijn de Tecmar Helical-Scan 2200 en de Plus Impulse spectaculair. De Helical-Scan is een externe SCSI tapedrive met een opslagcapaciteit van 2,2 Gigabyte. De Plus Impulse is een modulair harddiskconcept, met modules van 40 of 80 MB waarvan er een aantal aan elkaar te koppelen is tot een capaciteit van 22,6 Gigabyte. De drive heeft een toegangstijd van 12 miliseconde en een data-overdracht van 4 Mb/sec. De controller is geïntegreerd.

De Intel Connection Coprocessor is een multifunctioneel communicatieboard dat kan zorgen voor het in de achtergrond verzenden van faxen, bestanden en E-mail. positronika levert ook Kurzweil produkten, RasterOps randapparatuur voor de Macintosh, DCA netwerking hard- en software en er worden nieuwe produkten van 3COM getoond. Zo is de 3S/500 een nieuwe netwerkserver met standaard 2 of 8 MB RAM, diskdrives van 150-690 MB en een tape-eenheid van 250 MB of 2,3 GB. Uitbreidbaar tot 16 MB RAM en 6 GB aan harddisk.

Rank Xerox

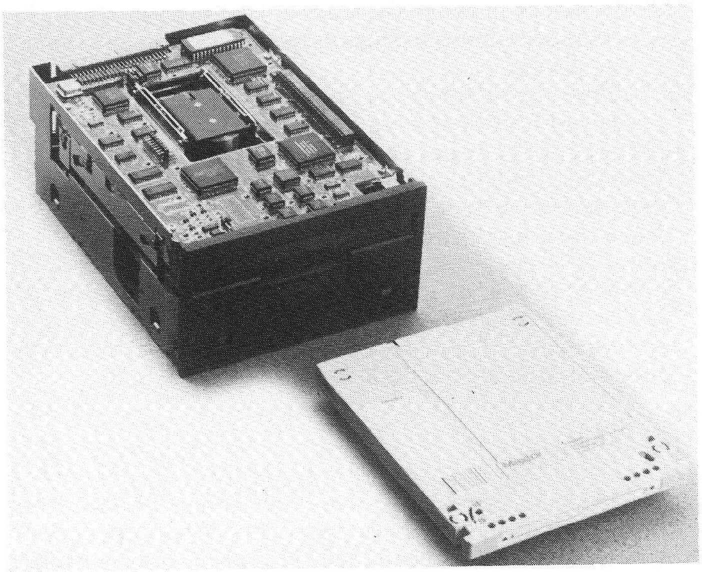
E 337 ***020 - 5106911***

De kernactiviteit van Rank Xerox is document processing en in dit kader worden enkele nieuwe produkten geïntroduceerd.

Er zijn nieuwe laserprinters: de Xerox 3700 release 5.0 (met release 5.0 printer software om de 9700 familie te emuleren), de Xerox 3700 voor A3 Graphics (voor afdrukken op A3 formaat), de Xerox 4030 en de Xerox 4650 en 4090 printers. De eerste zijn low-volume printers met nieuwe toepassingen ten opzichte van hun voorgangers. De 4650 en 4090 zijn high-volume printers met een resolutie van 600 x 600 dpi. De printers hebben een afdruksnelheid van resp. maximaal 50 en 92 pagina's per minuut.

- Het feit dat de meeste typen **uitsluitend data kunnen uitlezen**. Het wegschrijven geeft technische problemen. Toch komen er steeds meer optische drives op de markt die naast lezen ook data (soms slechts eenmalig) kunnen wegschrijven.

De verwachtingen rondom de educatieve en naslagmogelijkheden van optische schijfjes zijn hooggespannen. Omvangrijke handboeken passen nu op één CD schijf en het doorzoeken gaat in seconden. De *interactieve privéleraar* op de optische disk is al lang geen droom meer. Een PC wordt tot onderwijsstation, dat vragen stelt, de antwoorden met u doorneemt en fouten uitlegt. De Interactieve Video Techniek (**IVT**) zal de komende jaren een grote vlucht nemen. Hetzelfde geldt voor CD-informatiesystemen zoals VVV, musea, auto-navigatie, bibliotheken en encyclopedische informatie.



'Tahiti' 1 Gigabyte wisbare optische drive van Maxtor (foto Maxcom)

Waarom een extra drive?

Dankzij het bouwdoosconcept van de PC is meestal op uitbreiding met extra drives gerekend. In de grotere systeemkast zitten een drive chassis (drive bay), controllerkaart, verbindings- en voedingskabeltjes. U kunt er zo een diskettestation, harddisk of tapestreamer bijzetten. Maar waarom zou u dat eigenlijk doen? Daarvoor gelden in het algemeen de volgende practische argumenten:

- **Sneller en gemakkelijker werken**. Een machine met slechts één diskdrive is eigenlijk geen doen. Twee diskdrives zijn zelfs voor vele programma's een must. De voordelen van de harddisk zijn inmiddels al uit de doeken gedaan.
- Het door elkaar gebruiken van **verschillende disketteformaten**. Elk format zijn eigen drive.
- Het **vergroten van de opslagcapaciteit**. Met name bij de harddisks en de 1.2/1.44 MB-diskdrives.
- Het installeren van een **backup drive**. Tapestreamer, eventueel een extra harde schijf.

Een extra drive geeft vrijwel altijd meer gemak, datacapaciteit en vaak ook betere systeemprestaties.

Drives zijn onmisbaar voor uw computer. Alleen terminals kunnen zonder. Het werken met verschillende formats, grote data- opslagcapaciteit, een grotere efficiëncy plus betere systeemprestaties, snelle backups en IVT-educatie, het zou allemaal niet mogelijk zijn zonder (extra) drives. En om de kosten behoeft de PC gebruiker het tegenwoordig echt niet meer te laten.

U.S

Ondanks alle toekomstverwachtingen over het elektronische kantoor van morgen heeft het beeldscherm het papier nog altijd niet vervangen. Sterker nog, de sterke groei van het computerpark heeft een nog sterkere groei van de papierberg tot gevolg gehad. Printers vervullen nog steeds een belangrijke communicatiefunctie.

Printers

Papieren hardcopies



Beeldschermen zijn vooralsnog vermoeiende en lastig lezende dingen. Bovendien moet je dan eerst de computer en monitor weer aanschakelen om de magnetisch of optisch opgeslagen informatie te kunnen lezen. Even een papieren uitdraaitje uit een goede printer trekken gaat veel gemakkelijker en leest of corrigeert ook prettiger. Men noemt de uitgeprinte informatie de papieren **hardcopies** van de digitale data.

In de praktijk worden de papieren hardcopies oa. gebruikt voor het uitwisselen van teksten (steeds meer vormgegeven met behulp van Desktop Publishing), corrigeren van teksten, afdrukken van illustraties en het maken van technische ontwerpen of tekeningen.

Ook de opkomst van de fax heeft de papierberg doen groeien; de computer verzorgt de communicatie en de printer produceert de berichten.

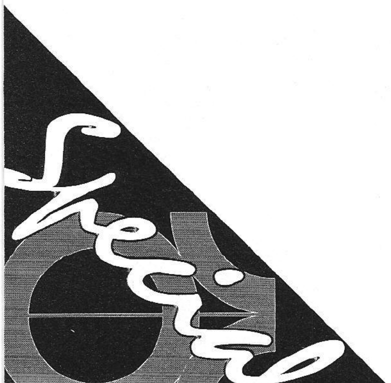
De mogelijkheden van de moderne printer zijn groot en het zal nog wel enkele decennia duren voordat alles via de HIRES of HDTV monitor verloopt.

Soorten printers

Er zijn vele typen printers in omloop die technisch gezien allemaal anders functioneren. Het oudste systeem werkt via een **mechanische overbrenging** van de inkt uit het inktlint op het papier. Dat kan met **letterhamertjes** bij de margrietwielprinter of computergestuurde schrijfmachine met roterende letterbol en via **naaldjes** bij de zogenaamde matrixprinters. De matrixprinters zijn tot op heden de meest gebruikte . **Plotters** zijn afdrukkers die de tekst en vooral grafische ontwerpen via viltstiftjes of pennetjes op het papier tekenen. Een speciale vorm zijn de electrostatische plotters, waarbij de inkt zich door ladingsverschillen aan de onderlaag (papier, sheets e.d.) hecht. **Thermische printers** hechten de inkt door lokale verhitting aan het papier. Daar komt verder geen enkel mechanisch geweld aan te pas. Deze printers worden voornamelijk bij draagbare modellen en videoprinters (ook in kleur) gebruikt.

Alle tot nu toe genoemde printers kenmerken zich door het feit dat de lettertekens of grafische symbolen stuk voor stuk, na elkaar en regel voor regel op het papier gezet worden. Dit in tegenstelling tot de zogenaamde **paginaprinters** (pageprinters) die in één run een complete pagina afdrukken. De **inkjetprinters** nemen een tussenpositie in. Deze apparaten spuiten de letters en grafische tekens op het papier. Sommige doen dat teken voor teken. Andere kunnen halve pagina's tegelijk bewerken.

De echte pageprinters zijn de **Laser**- en **LCD**-printers. Deze werken met licht om een gevoelige drum te activeren en functioneren verder als een soort fotokopieerapparaat. De laser- en LCD-printers worden ten gevolge van flinke prijsdalingen en hun bijna-drukwerkkwaliteit steeds populairder.



Matrixprinters

Dankzij de redelijke prijs (*f* 500,- tot *f* 3000,-) en een groot aantal mogelijkheden, vormen matrixprinters de eerste keuze waar een lichte geluidshinder, lagere afdruksnelheid en drukwerkkwaliteit geen voorname rol spelen. Ruwweg bepaalt het **aantal naaldjes**, doorgaans 9, 18 of 24, de uiteindelijke drukkwaliteit. Binnen dezelfde prijsklasse en gelijk aantal naalden kunnen echter grote verschillen optreden. Vaak is de keuze ook een kwestie van persoonlijke smaak. 24-Naalds matrixprinters geven een redelijke correspondentiekwaliteit. Sommige modellen hebben wat moeite met een echt diepzwart schrift.

Matrixprinters zijn, op de high-output modellen na, niet zo snel. De tekens worden stuk voor stuk en de pagina's regel voor regel afgedrukt. Het aantal tekens per seconde, **cps**, is een ruwe maat voor de snelheid. Dikwijls stelt de fabrikant de zaken gunstiger voor dan in de praktijk het geval is. Brede kolommen, vet- of schuinschrift en verschillende fonts vragen extra afdruktijd en dan lopen de prestaties snel terug. Bovendien geldt de opgegeven maximumsnelheid voor de lelijkste afdrukmodus, de beruchte **draft**-computerletter. In de correspondentiestand Near Letter Quality (**NLQ**) of Letter Quality (**LQ**) valt de snelheid terug tot 20 à 30% van het maximum.

De **drukkwaliteit** wordt opgegeven in het aantal puntjes (dots) per inch, de **DPI**. Normaal is de range 180-240 dpi. Geen fotokwaliteit, maar beter dan de krant.

Matrixprinters kunnen relatief veel. Het afdrukken van vreemde tekens, tekst en grafieken en verschillende fonts vormen meestal geen probleem. Ook kan de moderne matrixprinter tal van afdrukstandaards nabootsen (emuleren). Toch ontstaan regelmatig aansturingsproblemen door slecht werkende **drivers** (stuurprogramma's) of een verkeerde instelling. Een ander bezwaar vormt de lawaaiproduktie.

Letterwiel printers

De **margrietwiel**- en **schrijfmachineprinters** vormen een uitstervend ras. De schrijfmachine is qua kwaliteit nog onovertroffen. Helaas geldt dit ook voor de grote lawaaiproduktie en zeer lage afdruksnelheid. De keuze is beperkt en de prijzen zijn stevig. Toch kiezen conservatieve brieventikkers nogal eens voor deze printercategorie.

Thermische printers

De grote voordelen van **thermische printers** zijn de compacte bouw en de mogelijkheid tot goedkope kleurendruk. Thermische printers worden wel samen met portable computers verkocht. Een recentere toepassing is de videoprinter waarmee het schermbeeld in full colour kunt afgedrukt kan worden.

Bezwaren van thermische printers zijn dikwijls de enigszins tegenvallende afdrukkwaliteit en de papierprijs. De printers zelf zijn relatief goedkoop. Qua snelheid scoren zij gemiddeld.

Plotters

De **plotter** was vroeger een specialistisch apparaat voor de technisch tekenaar en industrieel ontwerper. Dat geldt nog altijd voor de peperdure topmodellen die vooral om hun nauwkeurigheid en perfecte lijnen gekocht worden. Behalve deze high-end markt voor CAD/CAM bestaat er ook een budget- en low-end markt. Reklametekenaars en desktop publishers grijpen ook meer en meer naar de plotter voor het maken van grafische ontwerpen. De amateur kan experimenteren met goedkope versies van circa *f* 500,-. Plotters nemen echt de tijd voor hun tekenwerk. Alleen de superdure elektrostatische modellen zijn wel redelijk snel. Een tweede probleem vormt de aansturing. Bij nogal wat software werd niet aan plotterbesturing gedacht.

Remidex

E 330 ***075 - 515615***

Van de bekende printerleverancier Nakajima laat Remidex een tweetal nieuwe modellen zien ter vervanging van de AR-50 en AR-55. De dot-matrix printers zijn 24-naalds printers, resp. een 80 en 132 koloms versie. De topsnelheid van beide types is met 12 cpi een 300 tekens per sekonde.

Rocomp

Z 469 ***040 - 425025***

Drie nieuwe computers van Cordata zijn op de Rocomp stand te zien. De CS4500 is een 12 MHz 80286 AT met 640 KB aan RAM op het moederbord. Een 3,5" 1.44 MB diskdrive is ingebouwd en een 20 of 40 MB harddisk is optioneel.

De CS4600 heeft dezelfde processor en hoeveelheid geheugen, dat op het moederbord uitgebreid kan worden tot 4 MB. Een 5,25" diskdrive is aanwezig. Deze AT is ondergebracht in een full size kast, in tegenstelling tot de CS4500 die in een kleine kast met lage ombouw is behuisd.

De CS5000-25 heeft een 80386-25 MHz, zero waitstate processor en biedt plaats aan 16 MB RAM op het moederbord. Standaard zit er 2 MB in.

Sirex Europe

O 110 ***053 - 333600***

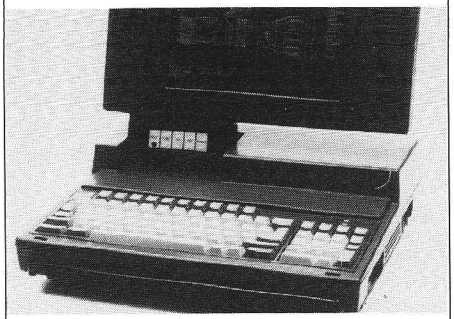
Sirex komt behalve met een nieuwe versie van de 'geluidchip' ook met een superkleine AT, die in Twente wordt geproduceerd. De Sirex 286T Stationmaster (12 MHz en 10 MHz uitvoeringen) is een complete AT met 3,5 inch drive en maximaal 200 MB harddisk en 4 MB geheugen maximaal op het moederbord (SIMM's). Er is een 8-bits en een 16-bits slot vrij.

Switch Computers

W 436A ***05410 - 17989***

Switch uit Oldenzaal brengt, naast de Rabbit computerserie, een nieuwe laptop. De LT 3400 is uitgerust met een Harris CPU 80C286-16. In de turbo mode is hij op te voeren tot 21.4 MHz. Het 1 MB geheugen is

uit te breiden tot 2, 4, of 5 MB. Het plasma scherm is MDA/CGA/ EGA compatible en levert een resolutie van 640 x 400. Ook kan er gelijktijdig een externe monitor op aangesloten worden.



Tracor Europa

A 222 *030 - 280855*

Van C-Itoh zijn er op de stand van Tracor nieuwe grafische terminals te zien voor onder andere DEC. De CIT 334 zwart wit terminal heeft een hoge resolutie van 800x500 met een 14 inch diagonaal. Er zijn diverse emulaties beschikbaar waarbij er sessies van twee verschillende systemen gelijktijdig in beeld te brengen zijn.

Tradecom International

D 721 *01714 - 14300*

Tradecom richt haar stand in met produkten in de sfeer van scanning en vectorisatie (CAD), optische systemen en datacommunicatie. Op het gebied van CAD worden complete systemen getoond en nieuwe versies van Rocad en Robuild.

Het optische system DocMan heeft sinds vorig jaar een aantal uitbreidingen ondergaan en het netwerksysteem van TCI verbindt beide reeksen produkten.

Xios Systems

R 807 *015 - 619561*

Xios uit Delft legt op de beurs de nadruk op het nieuwe Xios 7400 werkstation, de nieuwe Xios PC-lijn en nieuwe mogelijkheden van documentconversie.

Met de Xios 7400 is 'Mode Switching' tussen de Xios, Dos en Unix omgevingen mogelijk. Het systeem heeft een Xios tekstverwerker, window omgevingen, interactieve printer support, 3270 SNA Gateway, TCP/IP, LAN en WAN koppelingen, etc.

Inkjet printers

De inktspuiters stonden in een kwaad daglicht vanwege allerlei geknoei met verstopte inktbuisjes waar geen druppel meer uitkwam. Dankzij de geavanceerde en attractief geprijsde bureaumodellen van HP lijkt de **inkjetprinter** weer toekomst te krijgen. Deze apparaten benaderen in kwaliteit de laserprinter en kunnen tekst gecombineerd met grafieken afdrukken. Bij desktop publishing kan het een bezwaar zijn dat deze printers soms niet meer dan een halve pagina tegelijk kunnen afdrukken.

Of de inkjets een deel van de lasermarkt zullen wegkapen valt nog te bezien. De prijstelling is gunstig en zelfs lager dan bij menige 24-naalds matrixprinter. Een ander voordeel is het muisstille afdrukken. Er zijn echter wel een paar beperkingen in snelheid en pagina omvang.

Laser- en LCD-printers

Wie snelheid en afdrukkwaliteit hoog in het vaandel voert komt bij de laser- of LCD- pageprinters terecht. Deze printers leveren **semidrukwerk** (300, 400, 600 of 800 dpi) af en hun **snelheid** wordt opgegeven in het aantal pagina's per minuut in plaats van cps. Desktop publishing, perfect ogende correspondentie, het maken van formulieren e.d., het zou allemaal veel minder zijn zonder deze moderne kantoorprinters. De kantooromgeving vaart wel bij de geringe geluidsproductie.

De **laser** bevat relatief veel mechanische onderdelen, wat een bron van storingen kan vormen. De nieuwere LCD modellen (Liquid Cristal Display-sluiter i.p.v. een schrijvende laserstraal) zouden in deze wat betrouwbaarder zijn. In het verleden waren er nog wat andere problemen. De toner gaf veel troep, de machines vervuilden snel en men kon met een beetje pech de monteur aan het apparaat vastketenen. Ook vielen de aanschaf- en onderhoudsprijzen flink tegen. Een laservelletje kostte al gauw zo'n 50 cent. Verder was het werken met enveloppen en papierformaten groter dan A4 lastig.

De nieuwe generatie laser- en LCD-printers lijkt de meeste kinderziekten te hebben overwonnen.

Printer aankoop

De belangrijkste richtlijnen die u bij aankoop van een prinster in het achterhoofd dient te houden zijn: **afdrukkwaliteit**; **snelheid**; **compatibiliteit** met de hardware (parallelle of seriële aansturig, controllers); **degelijkheid**; **gebruiksvriendelijkheid**; kan uw **software de printer aansturen** en zo ja komen dan ook alle gewenste tekens er uit; heeft de printer voldoende **RAM/buffergeheugen** voor lange teksten, verschillende fonts, DTP, etc.; werkelijke **bedrijfskosten** (denk aan papier, inktlinten, tonercassettes, OPC-drums, ontwikkelunits, service, etc.); welke **papierformaten** kan de printer aan; hoe is de **lawaaiproductie**? zijn er ook **extra accessoires/uitbreidingen** leverbaar (oa. geheugenuitbreiding, besturingskaarten en fontcartridges); Welke **aanstuurtaal** (HP Laserjet, Postscript) wordt er gebruikt. Dit is belangrijk voor laserprinters en DTP toepassingen.

Gezien het gebruik dat u van de printer zult maken is het geen overbodige luze om deze zaken op een rijtje te zetten.

U.S.

```
990     data 5,212,169,90,141,6,212,238,0,
        207,238,0,207,173,0,207
995     data 201,78,208,5,169,0,141,0,207,
        174,0,207,189,193,16,188
1000    data 194,16,141,1,212,140,0,212,16
        9,17,141,4,212,208,179,234
1005    data 21,154,25,177,28,214,32,94,34
        ,75,32,94,28,214,22,227
1010    data 19,63,21,154,22,227,25,177,21
        ,154,21,154,20,100,21,154
1015    data 22,227,19,63,16,47,21,154,25,
        177,28,214,32,94,34,75
1020    data 32,94,28,214,22,227,19,63,21,
        154,22,227,25,177,22,227
1025    data 21,154,19,63,18,42,19,63,21,1
        54,21,154,21,154,173,0
1030    data 203,240,68,238,1,203,173,1,20
        3,201,30,208,58,169,0,141
1035    data 1,203,169,0,141,11,212,169,0,
        141,13,212,169,9,141,12
1040    data 212,238,2,203,238,2,203,173,2
        ,203,201,12,208,5,169,0
1045    data 141,2,203,174,2,203,189,89,17
        ,188,90,17,141,8,212,140
1050    data 7,212,169,33,141,11,212,96,9,
        247,9,104,10,143,10,143
1055    data 8,57,7,12,64
```

EINDE LISTING ufo-bas

Multiple-choice

Omdat wij weinig onderwijs programma's plaatsen schreef Obbe Vermeij het volgend programma voor ons. Multiple choice is een programma voor de C-64 waar mee multiple choice-testen gemaakt ge-edit en overhoord kunnen worden. Door de overzichtelijke structuur met veel REM-regels moet het ook zeer eenvoudig zijn om het programma naar eigen keus te veranderen of aan te vullen.

```
1000    rem      multiple choice (c-64)
1010    rem                door
1020    rem
1030    rem   **************************
1040    rem  **                          **
1050    rem **        obbe vermeij        **
1060    rem **                            **
1070    rem  **                          **
1080    rem   **************************
1090    rem
1100    rem         octopus software
1110    rem
1120    dim qu$(100),a1$(100),a2$(100),a3$
        (100),a4$(100),ri(100)
1130    gosub 4000:gosub 3880:gosub 3810
1140    :
1150    :
1160    rem ******** hoofd menu ********
1170    x=7
1180    m$=""
1190    b=1
1200    t$="hoofd[SPACE]menu"
1210    q$(1)="vragen[SPACE]overhoren"
1220    q$(2)="vragen[SPACE]bekijken/aanvu
        llen"
1230    q$(3)="nieuwe[SPACE]file[SPACE]ope
        nen"
1240    q$(4)="file[SPACE]saven"
1250    q$(5)="file[SPACE]laden"
1260    q$(6)="uitleg"
1270    q$(7)="uit[SPACE]het[SPACE]program
        ma"
1280    gosub4260
1290    on c goto 2610,1730,1480,3010,3200
        ,3390,1330
1300    :
1310    :
1320    rem ******* weg uit prg. *******
1330    m$="zeker[SPACE]weten[SPACE]?[SPAC
        E]u[SPACE]vernietigt[SPACE]de[SPAC
        E]file"
1340    t$=q$(7)
1350    q$(1)="nee,[SPACE]ik[SPACE]weet[SP
        ACE]het[SPACE]niet[SPACE]zeker"
1360    q$(2)="ik[SPACE]wil[SPACE]er[SPACE
        ]nog[SPACE]even[SPACE]over[SPACE]d
        enken"
1370    q$(3)="over[SPACE]5[SPACE]minuten,
        [SPACE]misschien"
1380    q$(4)="ja[SPACE],ik[SPACE]weet[SPA
        CE]het[SPACE]zeker[SPACE]!"
1390    x=4
1400    gosub 4260
1410    on c goto 1170,1170,1170,1420
1420    print"[SHIFT-CLR][7xSPACE]eindelij
        k[SPACE]bent[SPACE]u[SPACE]eruit[S
        PACE]!"
1430    poke 53269,0
1440    end
1450    :
1460    :
1470    rem ***** nieuwe file openen *****
1480    x=2
1490    t$="dit[SPACE]zal[SPACE]de[SPACE]o
        ude[SPACE]file[SPACE]vernietigen"
1500    m$="zeker[SPACE]weten[SPACE]?"
1510    q$(2)="ja[SPACE],ik[SPACE]wil[SPAC
        E]een[SPACE]nieuwe[SPACE]file."
1520    q$(1)="nee[SPACE],ik[SPACE]wil[SPA
        CE]de[SPACE]oude[SPACE]houden."
1530    gosub 4260
1540    on c goto 1170,1550
1550    print"een[SPACE]momentje[SPACE]a.u
        .b."
1560    poke 53269,0
1570    nq=0
1580    for t=0to 100:qu$(t)="":nextt
1590    for t=0to 100:a1$(t)="":nextt
1600    for t=0to 100:a2$(t)="":nextt
1610    for t=0to 100:a3$(t)="":nextt
1620    for t=0to 100:a4$(t)="":nextt
1630    print"[SHIFT-CLR][4xCRSR-DOWN][10x
        SPACE]wat[SPACE]zal[SPACE]de[SPACE
        ]nieuwe"
1640    input"naam[SPACE]van[SPACE]de[SPAC
        E]file[SPACE]zijn[2xSPACE](max.[SP
        ACE]40[SPACE]char.)[CTRL-2]";qu$(0
        )
1650    if len(qu$(0))>40 then goto  1630
1660    print"[SHIFT-CLR][3xCRSR-DOWN]vrag
        en[SPACE]maken[SPACE]door[SPACE]'v
        ragen[SPACE]bekijken/aan-[SPACE]vu
```

```
     llen'[2xSPACE]te[SPACE]kiezen."
1670 print"[6xCRSR-DOWN][6xSPACE]<spati
     e>[SPACE]voor[SPACE]doorgaan."
1680 gosub 4180
1690 goto 1170
1700 :
1710 :
1720 rem **** vragen bekijken/aanv.****
1730 t$=q$(2)
1740 m$="wat[SPACE]wilt[SPACE]u[SPACE]?
     "
1750 x=2
1760 q$(1)="oude[SPACE]vragen[SPACE]zie
     n[SPACE]of[SPACE]veranderen."
1770 q$(2)="nieuwe[SPACE]vragen[SPACE]i
     nvoeren."
1780 gosub 4260
1790 poke 53269,0
1800 on c goto 2170,1810
1810 print"[CTRL-2][SHIFT-CLR][2xCRSR-D
     OWN][8xSPACE]nieuwe[SPACE]vragen[S
     PACE]invoeren[CRSR-DOWN]"
1820 poke 646,12
1830 print"[5xSPACE]de[SPACE]file[SPACE
     ]bevat[2xSPACE]";nq;"[SPACE]vragen
     "
1840 print"[2xCRSR-DOWN][3xSPACE]voer[S
     PACE]xxx[SPACE]in[SPACE]voor[SPACE
     ]hoofd[SPACE]menu[3xSPACE]of[SPACE
     ]"
1850 gosub 1870
1860 goto 2020
1870 input"[CRSR-DOWN][SPACE]voer[SPACE
     ]volgende[SPACE]vraag[SPACE]in(max
     .[SPACE]38[SPACE]char)[SPACE][CTRL
     2]";qu$(nq+1)
1880 if len(qu$(nq+1))>38 then goto 187
     0
1890 if left$(qu$(nq+1),3)="xxx"then go
     to 1170
1900 poke 646,12: input"[SPACE]voer[2xS
     PACE]1e[SPACE]mogelijkheid[SPACE]i
     n[SPACE](max.31[SPACE]le.)[CTRL-2]
     ";a1$(nq+1)
1910 if len(a1$(nq+1))>31 then goto 190
     0
1920 poke 646,12: input"[SPACE]voer[SPA
     CE]2de[SPACE]mogelijkheid[SPACE]in
     [SPACE](max.31[SPACE]le.)[CTRL-2]"
     ;a2$(nq+1)
1930 if len(a2$(nq+1))>31 then goto 192
     0
1940 poke 646,12: input"[SPACE]voer[SPA
     CE]3de[SPACE]mogelijkheid[SPACE]in
     [SPACE](max.31[SPACE]le.)[CTRL-2]"
     ;a3$(nq+1)
1950 if len(a3$(nq+1))>31 then goto 194
     0
1960 poke 646,12: input"[SPACE]voer[SPA
     CE]4de[SPACE]mogelijkheid[SPACE]in
     [SPACE](max.31[SPACE]le.)[CTRL-2]"
     ;a4$(nq+1)
1970 if len(a4$(nq+1))>31 then goto 196
     0
1980 poke 646,12:input"[SPACE]wat[SPACE
     ]is[SPACE]het[SPACE]juiste[SPACE]a
     ntwoord[3xSPACE](1-4)[CTRL-2]";ri(
     nq+1)
1990 if ri(nq+1)>4 then goto 1980
2000 if ri(nq+1)<1 then goto 1980
2010 return
2020 poke 646,12
2030 print"[SHIFT-CLR][CRSR-DOWN][6xSPA
     CE]is[SPACE]het[SPACE]zo[SPACE]naa
     r[SPACE]uw[SPACE]zin[SPACE]?[2xCRS
     R-DOWN]"
2040 z=nq+1
2050 gosub 4580
2060 poke 646,15
2070 print"[3xCRSR-DOWN][3xSPACE]<f1>=[
     SPACE]correct[3xSPACE]<f3>=[SPACE]
     niet[SPACE]correct"
2080 get a$
2090 if a$="[F1]"then goto 2120
2100 if a$="[F3]"then goto 1810
2110 goto 2080
2120 nq=nq+1
2130 goto 1810
2140 :
2150 :
2160 rem *oude vragen veranderen/bek.**
2170 z=1:if nq=0then goto 1170
2180 poke 646,12
2190 print"[SHIFT-CLR][CRSR-DOWN][7xSPA
     CE]de[SPACE]file[SPACE]bevat[2xSPA
     CE]";nq;"[SPACE]vragen"
2200 print"[5xSPACE]u[SPACE]ziet[SPACE]
     nu[SPACE]vraag[SPACE]nummer[SPACE]
     ";z;"[2xCRSR-DOWN]":poke 646,15
2210 gosub 4580
2220 poke 646,12
2230 print"[SPACE]<f1>=zie[SPACE]volgen
     de[SPACE]vraag"
2240 print"[SPACE]<f3>=deze[SPACE]vraag
     [SPACE]veranderen"
2250 print"[SPACE]<f7>=zie[SPACE]vorige
     [SPACE]vraag"
2260 print"[SPACE]<space>=hoofd[SPACE]m
     enu[CRSR-UP]"
2270 get a$
2280 if a$="[F1]" then goto 2330
2290 if a$="[F3]" then goto 2370
2300 if a$="[F7]" then goto 2350
2310 if a$="[SPACE]" then goto 1170
2320 goto 2270
2330 if z=nq then goto 2270
2340 z=z+1:goto 2180
2350 if z=1 then goto 2270
2360 z=z-1:goto 2180
2370 t$="vraag[SPACE]veranderen"
2380 m$="wat[SPACE]wilt[SPACE]u[SPACE]v
     eranderen[SPACE]?"
2390 x=2
2400 q$(1)="iets[SPACE]aan[SPACE]de[SPA
     CE]vraag."
2410 q$(2)="ik[SPACE]wil[SPACE]de[SPACE
     ]vraag[SPACE]verwijderen."
2420 gosub 4260
2430 on c goto 2540,2440
2440 for t=z to 99
2450 qu$(t)=qu$(t+1)
2460 a1$(t)=a1$(t+1)
2470 a2$(t)=a2$(t+1)
2480 a3$(t)=a3$(t+1)
2490 a4$(t)=a4$(t+1)
2500 ri(t)=ri(t+1)
```

```
2510 nextt
2520 nq=nq-1:poke 53269,0
2530 goto 2170
2540 o=nq:nq=z-1
2550 poke 53269,0:print"[SHIFT-CLR]":go
     sub 1870
2560 z=nq+1:nq=o
2570 goto 2170
2580 :
2590 :
2600 rem ****** vragen overhoren ******
2610 if nq=0then goto 1170
2620 x=2
2630 t$=q$(1)
2640 m$="hoe[SPACE]moeten[SPACE]ze[SPAC
     E]overhoord[SPACE]worden[SPACE]?"
2650 q$(1)="beginnen[SPACE]met[SPACE]1[
     SPACE]dan[SPACE]2,3,4[SPACE]enz."
2660 q$(2)="in[SPACE]willekeurige[SPACE
     ]volgorde."
2670 gosub 4260
2680 ra=0:wa=0
2690 w=c
2700 for d=1 to nq
2710 z=d
2720 if w=2 then z=1+int(rnd(1)*nq)
2730 t$=qu$(0)
2740 m$=qu$(z)
2750 q$(1)=a1$(z)
2760 q$(2)=a2$(z)
2770 q$(3)=a3$(z)
2780 q$(4)=a4$(z)
2790 q$(5)=""
2800 q$(6)="stop[SPACE]de[SPACE]overhor
     ing"
2810 x=6
2820 gosub 4260
2830 on c goto 2840,2840,2840,2840,2820
     ,2870
2840 if c=ri(z)then ra=ra+1
2850 if c<>ri(z)then wa=wa+1
2860 next d
2870 poke 53269,0
2880 print"[SHIFT-CLR][2xCRSR-DOWN]";qu
     $(0)
2890 print"[CRSR-DOWN][SPACE]resultaten
     "
2900 print"[2xCRSR-DOWN][CTRL-2][SPACE]
     de[SPACE]leerling[SPACE]maakte[SPA
     CE]";ra+wa;"[SPACE]vragen"
2910 print"[CRSR-DOWN]hij/zij[SPACE]ant
     woorde[SPACE]";ra;"[SPACE]vragen[S
     PACE]juist."
2920 if wa+ra=0then wa=1
2930 print"[3xCRSR-DOWN]dus[SPACE]maakt
     e[SPACE]zij/hij[SPACE]"int(ra/(wa+
     ra)*100);"[SPACE]%[SPACE]goed."
2940 poke 646,15
2950 print"[7xCRSR-DOWN][4xSPACE]<spati
     e>[SPACE]=[SPACE]terug[SPACE]naar[
     SPACE]hoofd[SPACE]menu"
2960 gosub 4180
2970 goto 1170
2980 :
2990 :
3000 rem ********* save optie  ********
3010 t$=q$(4)
3020 x=2
3030 m$="zeker[SPACE]weten[SPACE]?"
3040 q$(1)="ja"
3050 q$(2)="nee"
3060 gosub4260
3070 on c goto 3080,1170
3080 open1,1,1,"multiple[SPACE]ch.[SPAC
     E]*"
3090 print#1,nq:print#1,qu$(0)
3100 for t=1 to nq
3110 print#1,qu$(t):print#1,a1$(t):prin
     t#1,a2$(t):print#1,a3$(t):print#1,
     a4$(t)
3120 print#1,ri(t)
3130 nextt
3140 close 1
3150 gosub3800
3160 goto 1170
3170 :
3180 :
3190 rem ******** load optie ********
3200 t$=q$(5)
3210 m$="zeker[SPACE]weten[SPACE]?"
3220 x=2
3230 q$(1)="ja"
3240 q$(2)="nee"
3250 gosub 4260
3260 on c goto 3270,1170
3270 open 1,1,0,"multiple[SPACE]ch.[SPA
     CE]*"
3280 input#1,nq:input#1,qu$(0)
3290 for t=1 to nq
3300 input#1,qu$(t):input#1,a1$(t):inpu
     t#1,a2$(t):input#1,a3$(t):input#1,
     a4$(t)
3310 input#1,ri(t)
3320 nextt
3330 close 1
3340 gosub3800
3350 goto 1170
3360 :
3370 :
3380 rem ******* instructies  ********
3390 v=1:poke53269,0
3400 poke 646,15:print"[SHIFT-CLR][CRSR
     -DOWN][SPACE]************[SPACE]in
     structies[SPACE]************"
3410 if v=0then v=1
3420 if v=4 then goto 1170
3430 print"[CRSR-DOWN][17xSPACE]-";v;"-
     [CTRL-2][CRSR-DOWN]"
3440 on v gosub 3490,3590,3670
3450 print"[2xCRSR-DOWN][4xSPACE]<spati
     e>[SPACE]=[SPACE]volgende[SPACE]bl
     adzijde"
3460 get a$
3470 if a$="[SPACE]"then v=v+1:goto 340
     0
3480 goto 3460
3490 print"[SPACE]multiple[SPACE]choice
     [SPACE]is[SPACE]bedoeld[SPACE]om[S
     PACE]leraren"
3500 print"[SPACE]te[SPACE]helpen.[SPAC
     E]met[2xSPACE]dit[2xSPACE]programm
     a[SPACE]is[SPACE]het"
3510 print"[SPACE]mogelijk[SPACE]om[SPA
     CE]een[SPACE]leerling[SPACE]te[SPA
     CE]overhoren"
3520 print"[SPACE]en[SPACE]de[SPACE]c-6
```

```
          4[SPACE]de[SPACE]score[SPACE]te[SP
          ACE]laten[SPACE]geven[SPACE]."
3530      print
3540      print"[SPACE]multiple[SPACE]choice
          [2xSPACE]kan[SPACE]ook[SPACE]handi
          g[2xSPACE]zijn"
3550      print"[SPACE]voor[SPACE]leerlingen
          .[SPACE]zij[SPACE]kunnen[SPACE]het
          [2xSPACE]pro-"
3560      print"[SPACE]gramma[SPACE]gebruike
          n[SPACE]om[SPACE]te[SPACE]oefenen.
          "
3570      print
3580      return
3590      print"[SPACE]multiple[SPACE]choice
          [SPACE]bevat[SPACE]geen[SPACE]save
          [SPACE]of"
3600      print"[SPACE]load[SPACE]optie[SPAC
          E]voor[2xSPACE]disk[SPACE]gebruike
          rs[2xSPACE]maar"
3610      print"[SPACE]als[SPACE]u[SPACE]een
          [SPACE]disk-drive[SPACE]bezit[SPAC
          E]kunt[SPACE]u[SPACE]zelf"
3620      print"[SPACE]zo'n[SPACE]routine[2x
          SPACE]schrijven.[SPACE]het[SPACE]p
          rogramma"
3630      print"[SPACE]is[SPACE]zeer[SPACE]o
          verzichtelijk[SPACE]door[SPACE]de[
          SPACE]rem-re-"
3640      print"[SPACE]gels.het[SPACE]is[SPA
          CE]vrij[SPACE]eenvoudig[SPACE]om[S
          PACE]multiple"
3650      print"[SPACE]choice[SPACE]naar[SPA
          CE]eigen[SPACE]wens[SPACE]aan[SPAC
          E]te[SPACE]passen."
3660      return
3670      print"[SPACE]er[SPACE]kan[SPACE]ma
          ar[SPACE]een[SPACE]'vragen-file'[S
          PACE]tegelijk"
3680      print"[SPACE]in[SPACE]de[2xSPACE]c
          omputer[SPACE]zijn.[SPACE]de[SPACE
          ]optie[SPACE]'nieuwe"
3690      print"[SPACE]file[SPACE]openen'[SP
          ACE]vernietigt[SPACE]dus[SPACE]van
          zelf[SPACE]de"
3700      print"[SPACE]oude[SPACE]file[SPACE
          ]![SPACE]![SPACE]!"
3710      print
3720      print"[SPACE]volgens[2xSPACE]mij[S
          PACE]is[2xSPACE]multiple[SPACE]cho
          ice[2xSPACE]vrij"
3730      print"[SPACE]eenvoudig[2xSPACE]te[
          SPACE]gebruiken[SPACE]en[SPACE]ik[
          SPACE]denk[SPACE]dat"
3740      print"[SPACE]u[2xSPACE]binnen[SPAC
          E]een[SPACE]half[SPACE]uur[2xSPACE
          ]in[SPACE]staat[SPACE]zult"
3750      print"[SPACE]zijn[SPACE]een[SPACE]
          multiple[SPACE]choice[SPACE]test[S
          PACE]te[SPACE]maken"
3760      print"[13xSPACE]succes[SPACE].[SPA
          CE].[SPACE]."
3770      return
3780      :
3790      :
3800      rem ******** int. sprite ********
3810      restore:for t=0to 63
3820      read a
3830      poke 832+t,a
3840      nextt
3850      return
3860      :
3870      :
3880      rem  sprite on screen + screencol.
3890      poke 53280,11:poke 53281,0
3900      poke 646,15:poke 53269,1
3910      poke 53248,57:poke 53249,119
3920      poke 53276,1:poke 53286,11
3930      poke 53285,12:poke 53287,15
3940      poke 2040,13
3950      poke 53270,peek(53270)and 239
3960      return
3970      :
3980      :
3990      rem ***** print intro *****
4000      poke 53280,1:poke 53281,1
4010      poke 650,128
4020      poke 53270,peek(53270)or 16
4030      poke 646,14:poke 53283,6
4040      poke 53282,14:poke 53269,0
4050      print"[SHIFT-CLR][5xCRSR-DOWN][2xS
          PACE]++++++++++++++++++++++++++++
          ++++++"
4060      print"[2xSPACE]++[31xSPACE]++"
4070      print"[2xSPACE]++[4xSPACE]multiple
          [SPACE]choice[12xSPACE]++"
4080      print"[2xSPACE]++[31xSPACE]++"
4090      print"[2xSPACE]++[13xSPACE]door[SP
          ACE]obbe[SPACE]vermeij[SPACE]++"
4100      print"[2xSPACE]++[31xSPACE]++"
4110      print"[2xSPACE]++++++++++++++++++
          ++++++++++++++++"
4120      print"[10xCRSR-DOWN][11xSPACE]typ[
          SPACE]spatie[SPACE]om[SPACE]verder
          [SPACE]te[SPACE]gaan"
4130      gosub 4170
4140      return
4150      :
4160      :
4170      rem ***** wacht op spatie *****
4180      get a$
4190      if a$<>"[SPACE]"then goto 4180
4200      return
4210      :
4220      :
4230      rem ****** menu maker ***********
4240      rem * x=aantal mogelijkheden    *
4250      rem *** q$(1-5)="mogelijheden" **
                    c=gekozen mogelijkheid(1
          -5)**
4260      gosub 3880
4270      poke 646,11
4280      print"[SHIFT-CLR][CRSR-DOWN][12xSP
          ACE]maak[SPACE]uw[SPACE]keuze";chr
          $(8);chr$(142)
4290      print"[SPACE]<f1>=op[2xSPACE]<f7>=
          neer[2xSPACE]<return>=bevestig[CRS
          R-DOWN]"
4300      poke 646,1
4310      print spc((40-len(t$))/2-1);t$;"[C
          RSR-DOWN]"
4320      poke 646,15
4330      print spc((40-len(m$))/2-1) ;m$;"[
          CRSR-DOWN]"
4340      for t=1 to x
4350      print"[CRSR-DOWN][8xSPACE]";q$(t)
```

```
4360   nextt
4370   c=1
4380   get a$
4390   if a$="[F7]"then goto 4490
4400   if a$="[F1]"then goto 4430
4410   if a$=chr$(13)then return
4420   goto 4380
4430   if c=1 then goto 4380
4440   c=c-1
4450   for a=1 to 16
4460   poke 53249,peek(53249)-1
4470   nexta
4480   goto 4380
4490   if c=x then goto 4380
4500   c=c+1
4510   for a=1 to 16
4520   poke 53249,peek(53249)+1
4530   nexta
4540   goto 4380
4550   :
4560   :
4570   rem ******** vraag tonen      *****
4580   printqu$(z)
4590   print"[2xCRSR-DOWN][SPACE]*[SPACE]
       ";a1$(z)
4600   print"[2xCRSR-DOWN][SPACE]*[SPACE]
       ";a2$(z)
4610   print"[2xCRSR-DOWN][SPACE]*[SPACE]
       ";a3$(z)
4620   print"[2xCRSR-DOWN][SPACE]*[SPACE]
       ";a4$(z)
4630   print"[2xCRSR-DOWN][3xSPACE]goede[
       SPACE]antwoord[SPACE]is[SPACE]nr.[
       SPACE]:[SPACE]";ri(z)
4640   return
4650   :
4660   :
4670   rem ****** sprite data   *******
4680   data0,32,0,0,40,0,0,42
4690   data0,170,166,128,234,165,160,245
4700   data85,104,245,85,90,245,85,124
4710   data255,245,240,255,247,192,0,63
4720   data0,0,60,0,0,48,0,0
4730   data0,0,0,0,0,0,0,0
4740   data0,0,0,0,0,0,0,0
4750   data0,0,0,0,0,0,0,0
```

EINDE LISTING mult-cho

REGEL 1000	209	REGEL 1550	118	REGEL 2100	102	REGEL 2650	41	REGEL 3200	37
REGEL 1010	195	REGEL 1560	252	REGEL 2110	83	REGEL 2660	158	REGEL 3210	170
REGEL 1020	143	REGEL 1570	129	REGEL 2120	203	REGEL 2670	89	REGEL 3220	60
REGEL 1030	211	REGEL 1580	155	REGEL 2130	83	REGEL 2680	41	REGEL 3230	120
REGEL 1040	55	REGEL 1590	103	REGEL 2140	58	REGEL 2690	76	REGEL 3240	198
REGEL 1050	97	REGEL 1600	104	REGEL 2150	58	REGEL 2700	235	REGEL 3250	89
REGEL 1060	55	REGEL 1610	105	REGEL 2160	22	REGEL 2710	80	REGEL 3260	30
REGEL 1070	154	REGEL 1620	106	REGEL 2170	124	REGEL 2720	226	REGEL 3270	72
REGEL 1080	211	REGEL 1630	221	REGEL 2180	198	REGEL 2730	117	REGEL 3280	230
REGEL 1090	143	REGEL 1640	79	REGEL 2190	184	REGEL 2740	152	REGEL 3290	251
REGEL 1100	39	REGEL 1650	249	REGEL 2200	143	REGEL 2750	234	REGEL 3300	174
REGEL 1110	143	REGEL 1660	233	REGEL 2210	94	REGEL 2760	236	REGEL 3310	33
REGEL 1120	113	REGEL 1670	66	REGEL 2220	198	REGEL 2770	238	REGEL 3320	214
REGEL 1130	126	REGEL 1680	90	REGEL 2230	184	REGEL 2780	240	REGEL 3330	209
REGEL 1140	58	REGEL 1690	82	REGEL 2240	144	REGEL 2790	241	REGEL 3340	88
REGEL 1150	58	REGEL 1700	58	REGEL 2250	54	REGEL 2800	196	REGEL 3350	82
REGEL 1160	212	REGEL 1710	58	REGEL 2260	54	REGEL 2810	64	REGEL 3360	58
REGEL 1170	65	REGEL 1720	104	REGEL 2270	6	REGEL 2820	89	REGEL 3370	58
REGEL 1180	103	REGEL 1730	34	REGEL 2280	99	REGEL 2830	14	REGEL 3380	98
REGEL 1190	37	REGEL 1740	39	REGEL 2290	104	REGEL 2840	32	REGEL 3390	111
REGEL 1200	19	REGEL 1750	60	REGEL 2300	104	REGEL 2850	220	REGEL 3400	209
REGEL 1210	104	REGEL 1760	192	REGEL 2310	223	REGEL 2860	198	REGEL 3410	163
REGEL 1220	201	REGEL 1770	18	REGEL 2320	84	REGEL 2870	252	REGEL 3420	192
REGEL 1230	161	REGEL 1780	89	REGEL 2330	49	REGEL 2880	24	REGEL 3430	110
REGEL 1240	141	REGEL 1790	252	REGEL 2340	207	REGEL 2890	245	REGEL 3440	61
REGEL 1250	117	REGEL 1800	29	REGEL 2350	195	REGEL 2900	228	REGEL 3450	89
REGEL 1260	188	REGEL 1810	158	REGEL 2360	208	REGEL 2910	101	REGEL 3460	6
REGEL 1270	108	REGEL 1820	198	REGEL 2370	201	REGEL 2920	100	REGEL 3470	80
REGEL 1280	89	REGEL 1830	20	REGEL 2380	17	REGEL 2930	64	REGEL 3480	86
REGEL 1290	229	REGEL 1840	90	REGEL 2390	60	REGEL 2940	201	REGEL 3490	36
REGEL 1300	58	REGEL 1850	93	REGEL 2400	26	REGEL 2950	31	REGEL 3500	74
REGEL 1310	58	REGEL 1860	77	REGEL 2410	219	REGEL 2960	90	REGEL 3510	70
REGEL 1320	199	REGEL 1870	53	REGEL 2420	89	REGEL 2970	82	REGEL 3520	136
REGEL 1330	169	REGEL 1880	80	REGEL 2430	30	REGEL 2980	58	REGEL 3530	153
REGEL 1340	39	REGEL 1890	143	REGEL 2440	247	REGEL 2990	58	REGEL 3540	157
REGEL 1350	76	REGEL 1900	167	REGEL 2450	107	REGEL 3000	9	REGEL 3550	241
REGEL 1360	8	REGEL 1910	15	REGEL 2460	3	REGEL 3010	36	REGEL 3560	145
REGEL 1370	79	REGEL 1920	237	REGEL 2470	5	REGEL 3020	60	REGEL 3570	153
REGEL 1380	243	REGEL 1930	18	REGEL 2480	7	REGEL 3030	170	REGEL 3580	142
REGEL 1390	62	REGEL 1940	239	REGEL 2490	9	REGEL 3040	120	REGEL 3590	73
REGEL 1400	89	REGEL 1950	21	REGEL 2500	13	REGEL 3050	198	REGEL 3600	3
REGEL 1410	3	REGEL 1960	241	REGEL 2510	214	REGEL 3060	89	REGEL 3610	2
REGEL 1420	39	REGEL 1970	24	REGEL 2520	2	REGEL 3070	29	REGEL 3620	130
REGEL 1430	252	REGEL 1980	253	REGEL 2530	83	REGEL 3080	73	REGEL 3630	110
REGEL 1440	128	REGEL 1990	216	REGEL 2540	97	REGEL 3090	14	REGEL 3640	156
REGEL 1450	58	REGEL 2000	215	REGEL 2550	61	REGEL 3100	251	REGEL 3650	176
REGEL 1460	58	REGEL 2010	142	REGEL 2560	96	REGEL 3110	18	REGEL 3660	142
REGEL 1470	229	REGEL 2020	198	REGEL 2570	83	REGEL 3120	53	REGEL 3670	244
REGEL 1480	60	REGEL 2030	199	REGEL 2580	58	REGEL 3130	214	REGEL 3680	211
REGEL 1490	76	REGEL 2040	134	REGEL 2590	58	REGEL 3140	209	REGEL 3690	117
REGEL 1500	170	REGEL 2050	94	REGEL 2600	2	REGEL 3150	88	REGEL 3700	141
REGEL 1510	24	REGEL 2060	201	REGEL 2610	5	REGEL 3160	82	REGEL 3710	153
REGEL 1520	24	REGEL 2070	194	REGEL 2620	60	REGEL 3170	58	REGEL 3720	201
REGEL 1530	89	REGEL 2080	6	REGEL 2630	33	REGEL 3180	58	REGEL 3730	218
REGEL 1540	29	REGEL 2090	96	REGEL 2640	112	REGEL 3190	208	REGEL 3740	62

REGEL 3750	76	REGEL 3950	37	REGEL 4150	58	REGEL 4350	67	REGEL 4550	58
REGEL 3760	45	REGEL 3960	142	REGEL 4160	58	REGEL 4360	214	REGEL 4560	58
REGEL 3770	142	REGEL 3970	58	REGEL 4170	15	REGEL 4370	38	REGEL 4570	166
REGEL 3780	58	REGEL 3980	58	REGEL 4180	6	REGEL 4380	6	REGEL 4580	14
REGEL 3790	58	REGEL 3990	76	REGEL 4190	149	REGEL 4390	111	REGEL 4590	165
REGEL 3800	31	REGEL 4000	39	REGEL 4200	142	REGEL 4400	102	REGEL 4600	166
REGEL 3810	138	REGEL 4010	249	REGEL 4210	58	REGEL 4410	83	REGEL 4610	167
REGEL 3820	200	REGEL 4020	289	REGEL 4220	58	REGEL 4420	88	REGEL 4620	168
REGEL 3830	159	REGEL 4030	0	REGEL 4230	254	REGEL 4430	176	REGEL 4630	246
REGEL 3840	214	REGEL 4040	98	REGEL 4240	223	REGEL 4440	20	REGEL 4640	142
REGEL 3850	142	REGEL 4050	166	REGEL 4250	221	REGEL 4450	176	REGEL 4650	58
REGEL 3860	58	REGEL 4060	137	REGEL 4260	96	REGEL 4460	192	REGEL 4660	58
REGEL 3870	58	REGEL 4070	160	REGEL 4270	197	REGEL 4470	195	REGEL 4670	162
REGEL 3880	186	REGEL 4080	137	REGEL 4280	64	REGEL 4480	88	REGEL 4680	214
REGEL 3890	87	REGEL 4090	231	REGEL 4290	9	REGEL 4490	215	REGEL 4690	30
REGEL 3900	0	REGEL 4100	137	REGEL 4300	148	REGEL 4500	19	REGEL 4700	201
REGEL 3910	212	REGEL 4110	190	REGEL 4310	82	REGEL 4510	176	REGEL 4710	242
REGEL 3920	98	REGEL 4120	94	REGEL 4320	201	REGEL 4520	191	REGEL 4720	169
REGEL 3930	153	REGEL 4130	89	REGEL 4330	68	REGEL 4530	195	REGEL 4730	55
REGEL 3940	237	REGEL 4140	142	REGEL 4340	180	REGEL 4540	88	REGEL 4740	55

PRINT OUT C-128 met Grafiek

Grafiek 128

Kyrill Poelmans uit Oegstgeest maakte een programma om grafieken te maken op de commodore 128. Een diskdrive is nodig om de grafieken op te slaan en natuurlijk is een printer ideaal om ze uit te kunnen printen De mogelijkheden van het programma zijn o.a. staaf grafiek, 3d staaf grafiek, taart grafiek, sinus grafiek, sinus sprite, save/laad/scratch/print/bekijk-grafiek. Eigenlijk te veel om op te noemen. Er zit een kleine handleiding bij waarin alles wordt uitgelegd. De delen van het programma die in eigen programma gebruikt kunnen worden staan achter in de handleiding. De print-routine is geschreven voor de General Electric THP 8100, maar werkt ook op de MPS 801/803. Als u een andere printer heeft kunt u misschien wat kleine wijzigingen aanbrengen.

```
10    rem*******************************
20    rem********grafiek maker 128******
30    rem**************door************
40    rem**********kyrill poelmans******
50    rem********voor commodore info****
60    rem*******************************
70    fast:a=0:b=0:f=0:p=0:dima(32):dimb
      (8):dimf(8):dimd(13)
80    graphic1,1:circle,11,11,10,10,,,,6
      0:shapea$,1,1,21,21:sprsavea$,1:re
      m sprite 1= 6-hoek
90    printchr$(27)+"m"+chr$(27)+"e"
100   trap820:color0,1:color4,1:color5,8
      :graphic0,1
110   char,11,1,"grafiek[SPACE]maker[SPA
      CE]c128":char,18,3,"menu":color5,4
120   char,5,6,"1.[SPACE]staafgrafiek[SP
      ACE](1-32[SPACE]staven)":color5,5
130   char,5,8,"2.[SPACE]3d[SPACE]staafg
      rafiek[SPACE](1-12[SPACE]staven)":
      color5,15
140   char,5,10,"3.[SPACE]taart-grafiek[
      SPACE](1-10[SPACE]segmenten)":colo
      r5,7
150   char,5,12,"4.[SPACE]sinus-grafiek"
      :color5,9
160   char,5,14,"5.[SPACE]sinus[SPACE]sp
      rite":color5,2
170   char,5,16,"6.[SPACE]save[SPACE]gra
      fiek":color5,6
180   char,5,18,"7.[SPACE]laad[SPACE]gra
      fiek":color5,10
190   char,5,20,"8.[SPACE]bekijk/print[S
      PACE]grafiek":color5,16
200   char,5,22,"9.[SPACE]scratch[SPACE]
      grafiek":color5,3
210   char,13,24,"maak[SPACE]uw[SPACE]ke
      uze!":slow
220   geta
230   onagoto250,360,510,620,680,830,860
      ,740,920
240   goto220
250   rem**staaf grafiek 1**
260   color5,8:graphic0,1:char,12,1,"sta
      af[SPACE]grafiek[SPACE]1":print
270   l=0:a=0:k=0:input"titel";w$:input"
      aantal[SPACE]staven[SPACE](1-32)";
      h:ifh>32then270
280   forl=1toh:graphic0,1:print"waarde[
      SPACE]staaf[SPACE]";l;"[SPACE]is";
290   inputa(l):ifa(l)<0ora(l)>175thenpr
      int"dat[SPACE]kan[SPACE]helaas[SPA
      CE]niet":goto290:elsenext:graphic1
      ,1:char,0,0,w$
300   draw,25,10to25,190to315,190
310   fori=10to190step10:draw,23,ito27,i
      :nexti
320   fori=33to315step9:draw,i,188toi,19
      2:nexti
330   forl=1toh:fork=atoa+6:draw,30+(k+1
      ),186to30+(k+1),186-a(l):nextk:a=a
      +9:nextl
340   getkeya$:goto100
350   rem*** 3d staaf grafiek ***
360   max=0:color5,8:graphic0,1:char,12,
      1,"3d[SPACE]staaf[SPACE]grafiek"
370   print:input"titel";u$:input"hoevee
      l[SPACE]staven(1-12)";d
380   ifd<1ord>13then100
390   fori=1tod:print"hoogte[SPACE]staaf
      ";i;:inputa(i):ifa(i)>maxthenmax=a
      (i)
400   nexti:print"kloppen[SPACE]de[SPACE
      ]gegevens[SPACE](j/n)?":getkeya$
410   ifa$<>"j"then360
420   color0,1:color4,1:graphic1,1:char,
      0,0,u$
430   draw,2,10to2,180to319,180:forw=175
      to10step-5:draw,0,wto4,w:nextw
```

```
440  forq=17to315step26:draw,q,178toq,1
     82:nextq
450  char,2,24,"j[2xSPACE]f[2xSPACE]m[2
     xSPACE]a[2xSPACE]m[3xSPACE]j[2xSPA
     CE]j[3xSPACE]a[2xSPACE]s[2xSPACE]o
     [2xSPACE]n[2xSPACE]d"
460  fori=1tod:a(i)=180-a(i)/max*165
470  box,i*26-20,a(i),i*26,175,0,1
480  draw,i*26-20,a(i)toi*26-14,a(i)-4t
     oi*26+4,a(i)-4toi*26+4,169toi*26,175
490  draw,i*26,a(i)toi*26+4,a(i)-4
500  nexti:getkeya$:goto100
510  rem*** taart grafiek ***
520  color5,8:graphic0,1:char,14,1,"taa
     rt[SPACE]grafiek"
530  print:input"titel";y$:p=0:n%=0:inp
     ut"[CRSR-DOWN]hoeveel[SPACE]segmen
     ten";n%:graphic0,1
540  forx=1ton%:print"waarde[SPACE]segm
     ent";x;:inputp(x)
550  t=t+p(x):nextx:forx=1ton%
560  p(x)=int(p(x)/(t/360)):next
570  rem** teken grafiek **
580  graphic1,1:color1,2:char,0,0,y$:x=
     160:y=100:r=90
590  forg=1ton%:ea=ea+p(g)
600  printba,ea:circle1,x,y,r,,ba,ea,0,1
610  xr=rdot(0):yr=rdot(1):draw1,xr,yrt
     ox,y:next:getkeya$:goto100
620  rem ** sinus-grafiek **
630  color5,8:graphic0,1:char,13,1,"sin
     us[SPACE]grafiek"
640  print:input"sinus[SPACE]waarde";g
650  print:input"hoogte[SPACE](1-99)";s
     :ifs>99then 650
660  graphic1,1:draw,0,100to320,100:for
     t=1to320:a=int(s*sin(t/g)):draw1,t
     ,100-a:next
670  getkeya$:goto100
680  rem** sprite sinus **
690  color5,8:graphic0,1:char,14,1,"sin
     us[SPACE]sprite"
700  graphic0,1:print:input"sinus[SPACE
     ]waarde";g
710  print:input"hoogte[SPACE](1-99)";s
     :ifs>99then 710
720  graphic1,1:fort=1to320:a=int(s*cos
     ((t/g))):movespr1,t,141-a:sprite1,
     1,7:next
730  getkeya$:sprite1,0,1:goto100
740  rem ** print grafiek **
750  print"[SHIFT-CLR]u[SPACE]ziet[SPAC
     E]zometeen[SPACE]de[SPACE]grafiek[
     SPACE]en[SPACE]dan[SPACE]is[SPACE]
     'p'[SPACE]voor[SPACE]printen[SPACE
     ]en[SPACE]elke[SPACE]andere[SPACE]
     toets[SPACE]voor[SPACE]menu"
760  char,0,24,"druk[SPACE]nu[SPACE]een
     [SPACE]toets",1:getkeya$
770  graphic1:getkeya$:ifa$<>"p"then 100
780  open4,4:fort=312to0step-8
790  draw1,t,0tot,319:shapep$,t,0,t+7,199
800  p$=left$(p$,200):print#4,chr$(8)+p$
810  next:print#4,chr$(15):close4:goto100
820  sprite1,0,1:open15,8,15,"i":close1
     5:goto100
830  scnclr:char,14,1,"save[SPACE]grafi
     ek":char,1,3,"geef[SPACE]naam[SPAC
     E]max[SPACE]13[SPACE]tekens"
840  inputg$
850  bsave"gm."+g$,p7168top16384:goto100
860  scnclr:char,12,1,"laad[SPACE]grafi
     ek"
870  char,1,3,"geef[SPACE]naam[SPACE]of
     [SPACE]$[SPACE]voor[SPACE]dir"
880  inputf$
890  iff$="$"then 900:else 910
900  directory"gm.*=p":getkeya$:goto860
910  bload"gm."+f$,p7168:goto100
920  scnclr:char,14,1,"scratch[SPACE]gr
     afiek"
930  char,1,3,"disk[SPACE]in[SPACE]driv
     e[SPACE]en[SPACE]toets":getkeya$:p
     rint
940  directory"gm.*=p"
950  print"welke[SPACE]grafiek[SPACE]mo
     et[SPACE]gescratched[SPACE]worden"
     :inputm$
960  scratch"gm."+m$:goto100
```

EINDE LISTING 128-graf

REGEL 10	35	REGEL 240	29	REGEL 470	137	REGEL 700	9	REGEL 930	118
REGEL 20	93	REGEL 250	208	REGEL 480	179	REGEL 710	139	REGEL 940	171
REGEL 30	175	REGEL 260	7	REGEL 490	58	REGEL 720	250	REGEL 950	159
REGEL 40	227	REGEL 270	172	REGEL 500	88	REGEL 730	124	REGEL 960	10
REGEL 50	28	REGEL 280	88	REGEL 510	0	REGEL 740	189		
REGEL 60	35	REGEL 290	232	REGEL 520	18	REGEL 750	93		
REGEL 70	54	REGEL 300	217	REGEL 530	230	REGEL 760	20		
REGEL 80	215	REGEL 310	209	REGEL 540	55	REGEL 770	162		
REGEL 90	179	REGEL 320	28	REGEL 550	30	REGEL 780	234		
REGEL 100	48	REGEL 330	208	REGEL 560	254	REGEL 790	81		
REGEL 110	235	REGEL 340	83	REGEL 570	167	REGEL 800	17		
REGEL 120	118	REGEL 350	106	REGEL 580	239	REGEL 810	148		
REGEL 130	31	REGEL 360	124	REGEL 590	76	REGEL 820	113		
REGEL 140	176	REGEL 370	63	REGEL 600	232	REGEL 830	22		
REGEL 150	21	REGEL 380	244	REGEL 610	93	REGEL 840	240		
REGEL 160	194	REGEL 390	139	REGEL 620	239	REGEL 850	201		
REGEL 170	136	REGEL 400	104	REGEL 630	39	REGEL 860	105		
REGEL 180	153	REGEL 410	34	REGEL 640	100	REGEL 870	30		
REGEL 190	243	REGEL 420	67	REGEL 650	142	REGEL 880	239		
REGEL 200	94	REGEL 430	204	REGEL 660	132	REGEL 890	248		
REGEL 210	59	REGEL 440	107	REGEL 670	83	REGEL 900	69		
REGEL 220	226	REGEL 450	180	REGEL 680	160	REGEL 910	207		
REGEL 230	38	REGEL 460	86	REGEL 690	6	REGEL 920	97		

PRINT OUT C-16 met o.a. Vijftigen

Checksum c16

```
10    rem ******************************
20    rem syntax.checksum
30    rem voor c-16 & plus/4
40    rem
50    rem syntax testen met 'sys 1536'
60    rem
70    rem v.851128.16        jan bodzinga
80    rem ******************************
90    i=1536        :rem beginadres
100   reada:ifa>=0then pokei,a:i=i+1:got
      o100
110   print"data[SPACE]is[SPACE]weggezet"
120   print"cheksum[SPACE]printen[SPACE]
      met[SPACE]'sys[SPACE]1536'
130   end
200   data 165, 43,166, 44,133
210   data  31,134, 32,169,147
220   data  32,210,255,160,  0
230   data 240,  3, 32, 73,  6
240   data  32, 73,  6,208,  1
250   data  96, 72,152, 32,131
260   data   6,168,104,234, 32
270   data  81,  6, 32, 73,  6
280   data 240, 12,201, 32,240
290   data 247, 24,101,252,133
300   data 252, 76, 37,  6,166
310   data 252,169,  0,132,253
320   data  32, 95,164,169, 13
330   data  32,210,255,164,253
340   data  76, 17,  6,200,208
350   data   2,230, 32,177, 31
360   data  96,162,  0,189,123
370   data   6,240,  6, 32,210
380   data 255,232,208,245, 32
390   data  73,  6,170, 32, 73
400   data   6,132,253, 32, 95
410   data 164,162,  3,169, 32
420   data  32,210,255,202,208
430   data 250,169,  0,133,252
440   data 164,253, 96, 82, 69
450   data  71, 69, 76, 32,  0
460   data   0, 72,138, 72, 32
470   data 225,255,240,251,104
480   data 170,104, 96, -1
```

** EINDE LISTING checks16

Checksum Checksum C-16

REGEL 10	249	REGEL 280	186
REGEL 20	247	REGEL 290	248
REGEL 30	121	REGEL 300	118
REGEL 40	143	REGEL 310	204
REGEL 50	75	REGEL 320	165
REGEL 60	143	REGEL 330	252
REGEL 70	8	REGEL 340	106
REGEL 80	249	REGEL 350	98
REGEL 90	103	REGEL 360	163
REGEL 100	2	REGEL 370	45
REGEL 110	245	REGEL 380	0
REGEL 120	237	REGEL 390	58
REGEL 130	128	REGEL 400	108
REGEL 200	210	REGEL 410	159
REGEL 210	208	REGEL 420	245
REGEL 220	142	REGEL 430	202
REGEL 230	1	REGEL 440	176
REGEL 240	3	REGEL 450	12
REGEL 250	157	REGEL 460	54
REGEL 260	155	REGEL 470	43
REGEL 270	215	REGEL 480	1

C-16 slang

Van Jos van Oijen ontvingen wij het programma Ruimteslang. De bedoeling van dit spelletje is dat je de slang door een veld probeert te loodsen dat doorsneden is door een onderbroken lijn. Het zal blijken dat dit veel moeilijker is dan het lijkt

```
1     rem ruimteslang ii / c16
2     rem
3     rem jos van oijen
4     rem
5     rem boven-leeuwen
6     rem
10    color0,1:color4,1:scnclr:printchr$
      (8)chr$(14):vol8:q=3292:n=1
20    char,8,0,"[CTRL-2]Jos[SPACE]van[SP
      ACE]Oijen[SPACE]presenteert":char,
      9,2,"[CTRL-7]***[SPACE]RUIMTE[SPA
      CE]SLANG[SPACE]II[SPACE]***"
30    fory=1to7:color1,8,y:print"[HOME][
      5xCRSR-DOWN]Na[SPACE]enkele[SPACE]
      jaren[SPACE]door[SPACE]het[SPACE]s
      terrenveld[SPACE]te"
40    print"hebben[SPACE]gezworven,[SPAC
      E]is[SPACE]de[SPACE]ruimteslang[SP
      ACE]nu"
50    print"aangekomen[SPACE]in[SPACE]ee
      n[SPACE]nieuw[SPACE]universum.[CRS
      R-DOWN]"
60    print"Door[SPACE]deze[SPACE]ervari
      ng[SPACE]heeft[SPACE]hij[SPACE]nu[
      SPACE]een"
70    print"aantal[SPACE]extra[SPACE]lev
      ens[SPACE](3)[SPACE]gekregen.[CRSR
      -DOWN]"
80    print"Bots[SPACE]niet[SPACE]tegen[
      SPACE]de[SPACE]lijn!!!!!!!!!!!!!!!
      !"
90    color1,3,7-y:char,3,15,"[CTRL-9]<[
      CTRL-0][SPACE]=[SPACE]links[5xSPAC
      E]c1989[5xSPACE][CTRL-9]>[CTRL-0][
      SPACE]=[SPACE]rechts":color1,11,y
100   char,6,19,"Zullen[SPACE]we[SPACE]n
      u[SPACE]weer[SPACE]beginnen?"
110   char,9,21,"Druk[SPACE]dan[SPACE]op
      [SPACE]een[SPACE]toets!":nexty
120   getg$:ifg$=""then30
130   gosub450:scnclr:char,9,12,"[CTRL-8
      ][CTRL-9]J[CTRL-0]oystick[SPACE]1[
      SPACE]of[SPACE][CTRL-9]K[CTRL-0]ey
      board?":e$="d":f$="6"
140    getg$:ifg$<>"j"andg$<>"k"then140
150   ifg$="k"thene$=",":f$="."
160   gosub450:color0,6,2
170   printchr$(142)"[SHIFT-CLR][CTRL-9]
      [CTRL-1][SPACE]ruimteslang[SPACE]i
      i[2xSPACE]S[3xSPACE]level"n"[CRSR-
      LEFT][SPACE]score[6xSPACE]":gosub4
      40
180   fora=3112to4032step40:pokea,160:po
      kea+39,160:pokea-1024,0:pokea-985,
      0:next
190   print"[CTRL-2]":poke2022,1:poke202
      3,1:poke2024,38
200   scnclr:z=1:x=1:l$="N":r$="M":z$=r$
210   x=x+z:ifx=38thenz=-1:z$=l$
```

```
220     ifx=1thenz=1:z$=r$
230     t=t+1:s=s+1:s$=str$(s):fore=2tolen
        (s$):poke3105+e,asc(mid$(s$,e,1))+
        128:next
240     ift=500then330
250     ifrnd(1)<.1thenprint:goto270
260     printtab(x)z$
270     geta$:ifa$=e$thenq=q-1
280     ifa$=f$thenq=q+1
290     p=peek(q):p1=peek(q-1):p2=peek(q+1
        ):p3=peek(q+40)
300     ifp=160thenq=qq:goto320
310      ifp=77orp=78or(p1<>32andp3<>32)or(
        p2<>32andp3<>32)then340
320     pokeq,81:qq=q:goto210
330     char,13,12,"[CTRL-4]gefeliciteerd!
        ":t=0:q=q+160:n=n+1:gosub380:gosub
        370:goto160
340     sound3,300,60:lv=lv-1:gosub440:pok
        e65299,60:forb=1to500:nextb
350     poke65299,208:pokeq,42:char,16,12,
        "[CTRL-1]jammer!!":iflv=-3then460
360     gosub370:goto190
370     char,9,16,"[COM-4]druk[SPACE]op[S
        PACE]de[SPACE]spatiebalk!":getkey
        g$:ifg$="[SPACE]"thengosub450:retu
        rn:else370
380     restore:fortt=1to44:reada,b:b=b/1.
        5
390     ifa=-1thenfora=1to150:nexta,tt
400     f=(111840.45/(1024-a))/4:aa=1024-(
        111840.45/f)
410     sound1,a,b:sound2,aa,b-1:next
420     ifq=4092thenchar,9,14,"[CTRL-4]je
        [SPACE]hebt[SPACE]alles[SPACE]geha
        ald!":goto490
430     return
440     poke3091,179+lv:return
450     fora=1020to800step-10:sound1,a,2:s
        ound1,a-114,3:next:return
460     char,11,14,"[CTRL-4]je[SPACE]leve
        ns[SPACE]zijn[SPACE]op!"
470     y=-1
480     y=y+1:ify>7then470:elsecolor1,3,y
490     char,6,16,"dit[SPACE]is[SPACE]het[
        SPACE]einde[SPACE]van[SPACE]de[SPA
        CE]reis":char,10,18,"nog[SPACE]een
        [SPACE]keer[SPACE](j/n)?"
500      getg$:ifg$<>"j"andg$<>"n"then480
510     ifg$="j"thenrun:elsesys62116
520     data785,15,-1,0,834,15,834,15,834,
        15,864,45,864,15,-1,0,904,15,904,1
        5,904,15
530     data929,45,929,15,-1,0,904,15,904,
        15,904,15,864,45,864,15,-1,0,834,1
        5,834,15
540     data834,15,929,30,929,15,917,15,-1
        ,0,897,15,897,15,897,15,864,45,864
        ,15,-1,0
550     data845,15,845,15,845,15,845,15,91
        7,30,929,15,904,30,904,30,904,30,9
        04,30,904,60
```

EINDE LISTING 16-slang

Checksum C-16 Slang

REGEL 1	69	REGEL 270	34
REGEL 2	143	REGEL 280	226
REGEL 3	213	REGEL 290	161
REGEL 4	110	REGEL 300	200
REGEL 5	174	REGEL 310	231
REGEL 6	143	REGEL 320	178
REGEL 10	173	REGEL 330	154
REGEL 20	119	REGEL 340	82
REGEL 30	129	REGEL 350	146
REGEL 40	135	REGEL 360	132
REGEL 50	183	REGEL 370	231
REGEL 60	217	REGEL 380	255
REGEL 70	234	REGEL 390	177
REGEL 80	254	REGEL 400	139
REGEL 90	98	REGEL 410	224
REGEL 100	153	REGEL 420	121
REGEL 110	9	REGEL 430	142
REGEL 120	60	REGEL 440	69
REGEL 130	31	REGEL 450	14
REGEL 140	119	REGEL 460	107
REGEL 150	49	REGEL 470	231
REGEL 160	55	REGEL 480	146
REGEL 170	243	REGEL 490	143
REGEL 180	208	REGEL 500	129
REGEL 190	251	REGEL 510	20
REGEL 200	149	REGEL 520	112
REGEL 210	9	REGEL 530	120
REGEL 220	138	REGEL 540	149
REGEL 230	173	REGEL 550	76
REGEL 240	99		
REGEL 250	118		
REGEL 260	59		

Vijftigen

Vijftigen is een kaartspel waarbij inzicht en geluk een grote rol spelen. De spelregels staan in het programma vermeld en zijn makkelijk te leren. De maker van dit programma is Jos van Oijen uit Boven-Leeuwen.

```
1       rem vijftigen / c16
2       rem
3       rem jos van oijen
4       rem
5       rem boven-leeuwen
6       rem
10      scnclr:color4,1:color0,1:printchr$
        (142)chr$(8)chr$(27)+"m":vol8:gosu
        b1120
20      char,3,2,"[CTRL-6]jos[SPACE]van[SP
        ACE]oijen[SPACE]presenteert[SPACE]
        :":poke2023,3:print"[2xCRSR-DOWN]"
30      kt$="UCCCI[CRSR-DOWN][5xCRSR-LEFT]
        B[3xSPACE]B[CRSR-DOWN][5xCRSR-LEFT
        ]B[3xSPACE]B[CRSR-DOWN][5xCRSR-LEF
        T]B[3xSPACE]B[CRSR-DOWN][5xCRSR-LE
        FT]B[3xSPACE]B[CRSR-DOWN][5xCRSR-L
        EFT]B[3xSPACE]B[CRSR-DOWN][5xCRSR-
        LEFT]JCCCK"
40      color1,7,7:print"O[8xCOM-Y]P[6xSPA
        CE]N[8xCOM-Y]M"
50      print"[COM-H][CTRL-9][8xSPACE][CTR
        L 0][COM-N][5xSPACE]N[SPACE][CTRL-
        9][SHIFT-|][6xSPACE][COM-*][2xCTRL
         0][SPACE]M"
60      color1,7,6:print"[COM-H][CTRL-9][S
        PACE][CTRL-0][SPACE][6xCOM-P][SHIF
        T `][5xSPACE][COM-H][CTRL-9][SHIFT
         |][CTRL-0][SHIFT-|][6xSPACE][COM-
        *][CTRL-9][COM-*][CTRL-0][COM-N]"
70      print"[COM-H][CTRL-9][SPACE][CTRL-
        0][SPACE][COM-H][11xSPACE][COM-H][
        CTRL-9][SPACE][CTRL-0][SPACE]N[4xC
```

```
        OM Y]M[SPACE][CTRL-9][SPACE][CTRL-
        0][COM-N]"
80      color1,7,5:print"[COM-H][CTRL-9][S
        PACE][CTRL-0][SPACE][COM-H][11xSPA
        CE][COM-H][CTRL-9][SPACE][CTRL-0][
        SPACE][COM-H][4xSPACE][COM-N][SPAC
        E][CTRL-9][SPACE][CTRL-0][COM-N]"
90      print"[COM-H][CTRL-9][SPACE][CTRL-
        0][SPACE]L[4xCOM-P][7xSPACE][COM-H
        ][CTRL-9][SPACE][CTRL-0][SPACE][CO
        M H][4xSPACE][COM-N][SPACE][CTRL-9
        ][SPACE][CTRL-0][COM-N]"
100     color1,7,4:print"[COM-H][CTRL-9][S
        PACE][CTRL-0][6xSPACE]M[6xSPACE][C
        OM H][CTRL-9][SPACE][CTRL-0][SPACE
        ][COM-H][4xSPACE][COM-N][SPACE][CT
        RL 9][SPACE][CTRL-0][COM-N]"
110     print"[COM-H][CTRL-9][6xSPACE][COM
         *][CTRL-0][SPACE]M[5xSPACE][COM-H
        ][CTRL-9][SPACE][CTRL-0][SPACE][CO
        M H][4xSPACE][COM-N][SPACE][CTRL-9
        ][SPACE][CTRL-0][COM-N]"
120     color1,7,3:print"L[5xCOM-P][SPACE]
        [COM-*][CTRL-9][COM-*][CTRL-0][COM
         N][5xSPACE][COM-H][CTRL-9][SPACE]
        [CTRL-0][SPACE][COM-H][4xSPACE][CO
        M N][SPACE][CTRL-9][SPACE][CTRL-0]
        [COM-N]"
130     print"[6xSPACE]M[SPACE][CTRL-9][SP
        ACE][CTRL-0][COM-N][5xSPACE][COM-H
        ][CTRL-9][SPACE][CTRL-0][SPACE][CO
        M H][4xSPACE][COM-N][SPACE][CTRL-9
        ][SPACE][CTRL-0][COM-N]"
140     color1,7,2:print"[6xSPACE][COM-N][
        SPACE][CTRL-9][SPACE][CTRL-0][COM-
        N][5xSPACE][COM-H][CTRL-9][SPACE][
        CTRL-0][SPACE][COM-H][4xSPACE][COM
         N][SPACE][CTRL-9][SPACE][CTRL-0][
        COM-N]"
150     print"[6xSPACE][COM-N][SPACE][CTRL
         9][SPACE][CTRL-0][COM-N][5xSPACE]
        [COM-H][CTRL-9][SPACE][CTRL-0][SPA
        CE][COM-H][4xSPACE][COM-N][SPACE][
        CTRL-9][SPACE][CTRL-0][COM-N]"
160     color1,7,1:print"[6xCOM-P]N[SPACE]
        [CTRL-9][SPACE][CTRL-0][COM-N][5xS
        PACE][COM-H][CTRL-9][SPACE][CTRL-0
        ][SPACE]M[4xCOM-P]N[SPACE][CTRL-9]
        [SPACE][CTRL-0][COM-N]"
170     print"[COM-H][6xSPACE][CTRL-9][SHI
        FT |][CTRL-0][SHIFT-|][COM-N][5xSP
        ACE][COM-H][COM-*][CTRL-9][COM-*][
        CTRL-0][6xSPACE][CTRL-9][SHIFT-|][
        CTRL-0][SHIFT-|][COM-N]"
180     color1,7,0:print"[COM-H][CTRL-9][6
        xSPACE][CTRL-0][SHIFT-|][SPACE]N[5
        xSPACE]M[SPACE][COM-*][CTRL-9][6xS
        PACE][CTRL-0][SHIFT-|][SPACE]N"
190     print"L[7xCOM-P]N[7xSPACE]M[8xCOM-
        P]N":char,33,17,"[CTRL-2]1989"
200     char,31,10,"[CTRL-9][SPACE][CTRL-0
        ][CRSR-DOWN][CRSR-LEFT]N":char,33,
        11,"UDI[COM-Q]DI":char,33,12,"[COM
         Q]CKB[SPACE]B":char,33,13,"JFKB[S
        PACE]B"
210     dimd$(38):q=6:a=rnd(-ti):a$="[SPAC
        E]0[SPACE]8[SPACE]910[SPACE]1[SPAC
        E]2[SPACE]3[SPACE]a":data"S","Z","
        A","X"
220     fora=1to4:reads$:forb=1to16step2:c
        $=mid$(a$,b,2):c=c+1:d$(c)=s$+c$:n
        extb,a
230     fora=1to32:x=32*rnd(1)+1:t$=d$(a):
        d$(a)=d$(x):d$(x)=t$:nexta
240     fora=33to38:d$(a)="-1":next:fora=1
        to3:c$(a)=d$(a):next:fora=4to6:s$(
        a-3)=d$(a):next
250     fora=800to1000step15:sound1,a,1:ne
        xt
260     char,11,22,"druk[SPACE]op[SPACE]e
        en[SPACE]toets!":getkeyg$:gosub11
        10:color1,1:color0,2,4:scnclr
270     ifpeek(213)=1then290
280     char,11,12,"[COM-2]instructies[SPA
        CE](j/n)?[CTRL-1]":getkeyg$:gosub1
        110:ifg$="j"thengosub1150
290     char,0,0,"[CTRL-9]":gosub1120:char
        ,14,11,"[CTRL-0][COM-7]1[4xSPACE]2
        [4xSPACE]3[CTRL-1]"
300     char,25,5,"[CTRL-6][COM-A]CCCCCCCC
        CCC[COM-S][CRSR-DOWN][13xCRSR-LEFT
        ]B[11xCRSR-RIGHT]B[CRSR-DOWN][13xC
        RSR-LEFT][COM-Z]CCCCCCCCCCC[COM-X]
        "
310     char,2,14,"[COM-A]CCCCCC[COM-S][CR
        SR-DOWN][8xCRSR-LEFT]B[6xCRSR-RIGH
        T]B[CRSR-DOWN][8xCRSR-LEFT][COM-Z]
        CCCCCC[COM-X]":gosub1020:goto570
320     rem kaart printen
330     char,p,z,kt$:kl$=left$(k$,1):ifkl$
        ="S"orkl$="Z"thenkt=3:elsekt=1
340     ifkl$="-"thenreturn
350     kr$=right$(k$,2):ifkr$="10"then410
360     kr$=right$(kr$,1)
370     ifkr$="0"thenkr$="7"
380     ifkr$="1"thenkr$="b"
390     ifkr$="2"thenkr$="v"
400     ifkr$="3"thenkr$="h"
410     char,p+1,z+1,kr$:color1,kt,3:char,
        p+3,z+1,kl$
420     char,p+1,z+5,kl$:color1,1:ifkr$<>"
        10"thenchar,p+2,z+5,"[SPACE]"+kr$:
        elsechar,p+2,z+5,kr$
430     return
440     rem teller
450     kr$=right$(k$,2):tt=val(kr$):ifkr$
        <>"[SPACE]a"then510
460     char,2,2,"[CTRL-2][CTRL-9]1[CTRL-0
        ][SPACE]een[SPACE]of[SPACE][CTRL-9
        ]2[CTRL-0][SPACE]elf?[CTRL-1]"
470      getkeyg$:ifg$<>"1"andg$<>"2"then47
        0
480     gosub1110:char,2,2,"[15xSPACE]"
490     ifg$="1"thentt=1:elsett=11
500     goto550
510     iftt<>10then550
520     char,2,2,"[CTRL-2][CTRL-9]+[CTRL-0
        ]tien[SPACE]of[SPACE][CTRL-9]-[CTR
        L 0]tien?[CTRL-1]"
530      tt=10:getkeyg$:ifg$<>"+"andg$<>"-"
        then530
540     gosub1110:char,2,2,"[15xSPACE]":if
        g$="-"thentt=-10:ift-10<0then520
550     t=t+tt:char,26,6,"aantal:"+str$(t)
        +"[SPACE]":return
560     rem begin
```

```
570    ifpeek(212)=1then750
580    ifpeek(212)=2then620
590    char,3,2,"wie[SPACE]begint[SPACE]e
       r,[SPACE]de[SPACE][CTRL-9]c[CTRL-0
       ]omputer[SPACE]of[SPACE][CTRL-9]j[
       CTRL-0]ij?"
600     getkeyg$:ifg$<>"c"andg$<>"j"then60
       0
610    gosub1110:char,3,2,"[34xSPACE]":if
       g$="c"then750
620    z=12:forp=12to22step5:k$=s$((p-7)/
       5):gosub330:next:v=1
630    rem speler aan beurt
640    ifright$(s$(1),2)=right$(s$(2),2)a
       ndright$(s$(1),2)=right$(s$(3),2)t
       hen990
650    char,3,15,"kaart?"
660    getkeyg$:k=val(g$):ifk<1ork>3then6
       60
670    gosub1110:ifs$(k)="-1"then660
680    k$=s$(k):p=17:z=3:gosub330:p=k*5+7
       :forz=13to17:char,p+1,z,"[3xSPACE]
       ":next
690    vl=0:gosub450:ifq<33thenq=q+1
700    s$(k)=d$(q):z=12:k$=s$(k):gosub330
710    ift>50then950
720    char,3,15,"[6xSPACE]"
730    rem
740    rem computer aan beurt
750    fora=0to1000:nexta
760    rem drie dezelfde
770    ifright$(c$(1),2)=right$(c$(2),2)a
       ndright$(c$(1),2)=right$(c$(3),2)t
       hen1090
780    rem sorteren
790    fora=1to3:kr$=right$(c$(a),2):tt=v
       al(kr$):ifkr$<>"[SPACE]a"then810
800    ift+11<51thentt=11:elsett=1
810    v(a)=tt:nexta
820    fora=1to3:e(a)=-2:b$(a)="":next:h=
       0
830    fora=1to3:forb=1to3:ifv(b)>e(a)the
       ne(a)=v(b):h=b:b$(a)=c$(b)
840    nextb:v(h)=-2:nexta:fora=1to3:v(a)
       =e(a):c$(a)=b$(a):next
850    rem weggooien
860    fora=3to1step-1:ifv(a)+t<51thenk=a
870    next
880    ifv(k)=10thenk=k+1:goto880
890    ifc$(3)="-1"thenk=2
900    ifc$(3)="-1"andc$(2)="-1"thenk=1
910    ifv(k)+t>50andv(1)=10thenk=1:v(1)=
       -10
920    k$=c$(k):p=17:z=3:gosub330:tt=v(k)
       :gosub550:ifq<33thenq=q+1
930    c$(k)=d$(q):sound1,300,3:ift>50the
       n990
940    ifv=1then640:else620
950    char,11,2,"[CTRL-8]je[SPACE]hebt[
       SPACE]verloren!!":poke212,1
960    c=peek(208):ifc<255thenpoke208,c+1
       :goto980
970    poke209,peek(209)+1:poke208,0
980    gosub1020:goto1040
990    char,11,2,"[CTRL-8]je[SPACE]hebt[
       SPACE]gewonnen!!":poke212,2
1000   v=peek(210):ifv<255thenpoke210,v+1
       :goto980
1010   poke211,peek(211)+1:poke210,0:goto
       980
1020   cc=peek(208)+256*peek(209):vv=peek
       (210)+256*peek(211)
1030   char,2,5,"[CTRL-4]computer:"+str$(
       cc):char,2,7,"[COM-8]speler[2xSPAC
       E]:"+str$(vv)+"[CTRL-1]":return
1040   a=0
1050   a=a+1:ifa=8thena=0
1060   color1,14,a:char,7,10,"wil[SPACE]j
       e[SPACE]nog[SPACE]een[SPACE]keer[S
       PACE](j/n)?":getg$
1070    ifg$<>"j"andg$<>"n"then1050
1080   gosub1110:ifg$="j"thenrun:elsefora
       =208to213:pokea,0:next:print"[SHIF
       T CLR][CTRL-1]doei!":end
1090   char,2,18,"[CTRL-2]drie[SPACE]goed
       ":char,2,20,"kijk[SPACE]maar[CTRL-
       1]"
1100   z=17:forp=12to22step5:k$=c$((p-7)/
       5):gosub330:next:goto950
1110   sound1,800,5:return
1120   char,0,0,"[CTRL-8]UCCCCCCCCCCCCCCC
       CCCCCCCCCCCCCCCCCCCCCCCI"
1130   fora=1to23:char,0,a,"B":char,39,a,
       "B":next
1140   char,0,24,"JCCCCCCCCCCCCCCCCCCCCC
       CCCCCCCCCCCCCCCCCK":return
1150   color4,6,3:color0,6,5:scnclr:print
       chr$(14):printtab(14)"INSTRUCTIES[
       CRSR-DOWN]"
1160   print"Er[SPACE]wordt[SPACE]om[SPAC
       E]de[SPACE]beurt[SPACE]een[SPACE]k
       aart[SPACE]gespeeld.[CRSR-DOWN]De[
       SPACE]waarde[SPACE]van[SPACE]deze[
       SPACE]kaart[SPACE]wordt[SPACE]bij[
       SPACE]";
1170   print"[SPACE]het[SPACE]totaal[SPAC
       E]opgeteld.[CRSR-DOWN]":print"Dege
       ne[SPACE]die[SPACE]de[SPACE]50[SPA
       CE]overschrijdt[SPACE]heeft[SPACE]
       ver-loren."
1180   print"[CRSR-DOWN]De[SPACE]waarde[S
       PACE]van[SPACE]de[SPACE]kaarten[SP
       ACE]is[SPACE]als[SPACE]volgt:[CRSR
       -DOWN]":print"AAS[3xSPACE]1[SPACE]
       of[SPACE]11"
1190   print"HEER[2xSPACE]3":print"VROUW[
       SPACE]2":print"BOER[2xSPACE]1":pri
       nt"10[4xSPACE]+10[SPACE]of[SPACE]-
       10":print"7[5xSPACE]0"
1200   print"De[SPACE]andere[SPACE]kaarte
       n[SPACE]hebben[SPACE]de[SPACE]waar
       de[SPACE]van[2xSPACE]het[SPACE]get
       al[SPACE]dat[SPACE]er[SPACE]opstaa
       t.[CRSR-DOWN]"
1210   print"Bij[SPACE]drie[SPACE]dezelfd
       e[SPACE]kaarten[SPACE]bv[SPACE]3[S
       PACE]azen[SPACE]heb[SPACE]je[SPACE
       ]ook[SPACE]gewonnen."
1220   getkeyg$:gosub1110:scnclr:printchr
       $(142):poke213,1:color4,1:color0,2
       ,4:return
```

EINDE LISTING 16-vijftigen

- PRINT OUT - PRINT OUT - PRINT OUT - PRINT OUT - PRINT

Checksum vijftigen

REGEL 1	14	REGEL 280	122	REGEL 610	67	REGEL 940	172
REGEL 2	143	REGEL 290	160	REGEL 620	137	REGEL 950	4
REGEL 3	213	REGEL 300	195	REGEL 630	172	REGEL 960	127
REGEL 4	110	REGEL 310	56	REGEL 640	174	REGEL 970	174
REGEL 5	174	REGEL 320	34	REGEL 650	243	REGEL 980	216
REGEL 6	143	REGEL 330	157	REGEL 660	211	REGEL 990	249
REGEL 10	174	REGEL 340	158	REGEL 670	191	REGEL 1000	170
REGEL 20	44	REGEL 350	115	REGEL 680	81	REGEL 1010	253
REGEL 30	222	REGEL 360	171	REGEL 690	233	REGEL 1020	94
REGEL 40	159	REGEL 370	7	REGEL 700	67	REGEL 1030	56
REGEL 50	90	REGEL 380	19	REGEL 710	58	REGEL 1040	35
REGEL 60	214	REGEL 390	40	REGEL 720	65	REGEL 1050	201
REGEL 70	6	REGEL 400	27	REGEL 730	143	REGEL 1060	139
REGEL 80	3	REGEL 410	188	REGEL 740	80	REGEL 1070	101
REGEL 90	193	REGEL 420	121	REGEL 750	6	REGEL 1080	168
REGEL 100	27	REGEL 430	142	REGEL 760	246	REGEL 1090	219
REGEL 110	229	REGEL 440	87	REGEL 770	150	REGEL 1100	108
REGEL 120	56	REGEL 450	113	REGEL 780	1	REGEL 1110	248
REGEL 130	252	REGEL 460	207	REGEL 790	180	REGEL 1120	54
REGEL 140	236	REGEL 470	68	REGEL 800	1	REGEL 1130	208
REGEL 150	217	REGEL 480	150	REGEL 810	63	REGEL 1140	141
REGEL 160	34	REGEL 490	26	REGEL 820	100	REGEL 1150	152
REGEL 170	231	REGEL 500	35	REGEL 830	214	REGEL 1160	219
REGEL 180	132	REGEL 510	57	REGEL 840	115	REGEL 1170	224
REGEL 190	227	REGEL 520	117	REGEL 850	51	REGEL 1180	6
REGEL 200	196	REGEL 530	43	REGEL 860	170	REGEL 1190	181
REGEL 210	163	REGEL 540	60	REGEL 870	130	REGEL 1200	122
REGEL 220	66	REGEL 550	224	REGEL 880	189	REGEL 1210	110
REGEL 230	226	REGEL 560	244	REGEL 890	160	REGEL 1220	12
REGEL 240	195	REGEL 570	89	REGEL 900	140		
REGEL 250	75	REGEL 580	86	REGEL 910	208		
REGEL 260	122	REGEL 590	95	REGEL 920	138		
REGEL 270	89	REGEL 600	105	REGEL 930	154		

Onze populaire informatie-rubriek voor de aktieve Geos-gebruiker

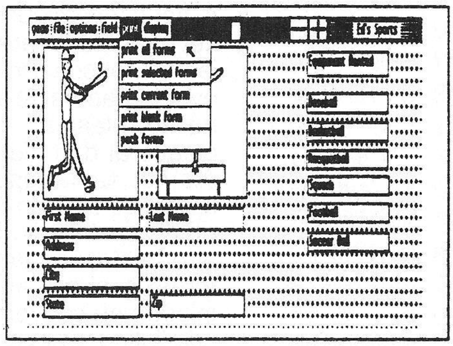
GEOS-INFO

Nederlandse handleiding GEOS 2.0

Een 'wanhopige' heer Werkhoven vraagt ons wanneer de Nederlandse handleiding van GEOS 2.0 op de markt verschijnt. Eindelijk is het dan zover. Reeds in het vorige nummer van Commodore Info van juli/aug. werd het boek **'GEOS 2.0, basishandleiding voor de aktieve gebruiker'** geïntroduceerd. Dit boek beschrijft niet alleen de werking van GEOS 2.0, maar ook die van GeoPaint, GeoWrite 2.1, GeoSpell, GeoMerge, Text Grabber en alle desk-accessoires. Verder behandelt het boek enkele applicaties, zoals GeoCalc, GeoFile, GeoPublish en de recent verschenen GeoChart. Echt een boek voor de ware 'GEOS-freak', zowel de beginnende als de gevordere gebruiker. Het boek, 144 pagina's dik en veel illustraties, is momenteel verkrijgbaar voor slechts fl. 34.95. U kunt het boek bestellen door dit bedrag (incl. verzendkosten) over te maken op gironummer 1585491 van **SAC Amsterdam** onder vermelding van **'GEOS basishandleiding'**

Printers?!!?

Bob Damman uit Hove (België) schrijft ons dat zijn STAR LC-10c niet de volle breedte van zijn GeoFile 128 wil afdrukken, maar slechts 3/4 deel. Dit probleem komt helaas wel vaker voor bij printers die afdrukken met een density van 60 dots per inch.



U zult bij het samenstellen van uw **forms** rekening moeten houden met de maximale printbreedte vn uw printer. Er is jammer genoeg geen andere oplossing te bedenken.

Reactie's van lezers

Van Patrick Cools uit Varsenare (België) ontvingen we naast een aantal bruikbare opmerkingen enkele nuttige Public Domain programma's. Allereerst reageert Patrick op een probleem uit Commodore Info nr. 2/89. Het betrof hier een gewiste **bootdisk**. Aldus luidt zijn advies, om altijd een copie van uw **bootdisk** achter de hand te houden. Deze disk zal dan niet als opstartdisk te gebruiken zijn, maar als uw originele **bootdisk** dan iets overkomt, copieert u de files van uw copie gewoon terug naar de originele **bootdisk**. Waar het hierbij om gaat is het identificatienummer van uw originele diskette. Er is echter één voorwaarde, op de **bootdiskette** mogen niet meer dan 35 tracks geformatteerd zijn. Voor vele GEOS gebruikers kan deze methode van het maken van een back-up een redmiddel zijn wanneer hun originele **bootdisk** in het ongerede raakt. Hartelijk dank voor de tip.

Voor al uw reacties, vragen en tips kunt u zich wenden tot:
GEOS Inforubriek
Postbus 43048
1009 ZA Amsterdam

Eindelijk begint het dan een beetje te komen, de Public Domain software voor gebruik in of met GEOS. Ditmaal een aantal nuttige programma's, waarmee u iconen van GEOS-files, plaatjes van PrintMaster en PrintShop kunt 'grabben' en naar een GeoPaint file kunt convertren.



Afb. 1. GeoPaint file

Plaatjes verzamelen voor GeoPaint

Met behulp van de in dit artikel behandelde programma's, welke zijn geprogrammeerd door Patrick Cools uit Varsenare, komt u in het bezit van een aantal converteerprogramma's, die plaatjes omzetten naar standaard graphics. Deze 'graphics schermen' kunnen vervolgens met behulp van het programma **GeoPaint Catcher** (gepubliceerd in Comm. Info nr. 1, jan/feb 1989) worden omgezet naar een GeoPaint file. Zie afbeelding 1.

Hoe gaat het in z'n werk? Allereerst moet u het machinetaal programma **'SYS POWER',** foutloos intikken en wegschrijven naar diskette. Dit programma wordt door de volgende programma's gebruikt voor het wissen van het grafische scherm, de kleuren en het laden van de tekeningen,

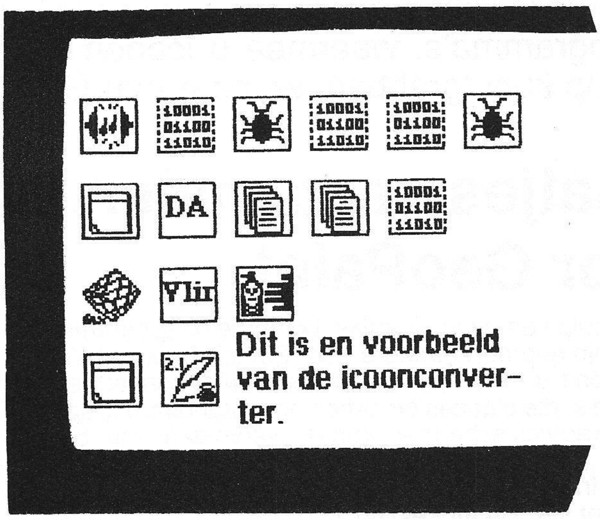
alsook het saven van het resultaat. Het programma moet worden ingevoerd met behulp van een machinetaal monitor en wordt weggeschreven onder de naam **"SYS POWER",033c,03fb**. Dit geheugengebied wordt normaal gebruikt door de cassettebuffer.

```
.:033C A0 80 8C 57 03 A9 60 8D
.:0344 4F 03 A9 00 A2 00 A0 00
.:034C 18 9D 00 00 E8 D0 FA EE
.:0354 4F 03 A0 00 CC 4F 03 D0
.:035C EF 60 A0 B1 CC 00 5C D0
.:0364 01 60 A9 5C 8D 4F 03 A9
.:036C 60 8D 57 03 A9 B1 4C 48
.:0374 03 A9 10 A6 33 A4 34 20
.:037C BD FF A2 08 A0 00 20 BA
.:0384 FF 60 20 75 03 A2 00 A0
.:038C 80 A9 00 20 D5 FF 4C A9
.:0394 03 20 75 03 A9 00 85 FB
.:039C A9 60 85 FC A9 FB A2 41
.:03A4 A0 7F 20 D8 FF 20 CC FF
.:03AC A9 01 20 C3 FF 60 EA 00
.:03B4 00 00 00 00 00 00 00 00
.:03BC 00 00 00 00 25 01 00 0A
.:03C4 00 00 00 00 01 00 3F 01
.:03CC 00 C7 60 C0 00 06 00 00
.:03D4 00 00 00 00 00 00 00 00
```

listing SYS POWER

Icoon 'grabber'

Het eerste programma, 'Converteren van GEOS iconen', is evenals de volgende programma's geschreven in BASIC. Het laadt GEOS file-iconen vanaf een GEOS diskette en toont deze op het scherm. Er kunnen maximaal 54 iconen op het scherm worden weergegeven, zie afb. 2.



afb. 2 GEOS iconen

Als u de gewenste iconen op het scherm heeft, kunt u het 'scherm' wegschrijven naar diskette. Het programma vraagt dan om een filenaam. Het file wordt weggeschrevn in 32 blokken. Naderhand kan men het file weer laden (om eventueel naar een GeoPaint file te convertren), door **LOAD"filenaam",8,1** in te voeren. Het scherm staat dan in geheugengebied \$6000 tot \$7f41. Vervolgens laadt u het programma GeoPaint Catcher, en selecteert het scherm, waarop u het weer wegschrijft naar diskette als een GeoPaint file. U kunt de iconen nu bewerken met GeoPaint.

Plaatjes converteren

De volgende twee programma's zijn interresant voor GEOS gebruikers, welke niet in het bezit zijn van de applicatie Graphics Grabber uit het pakket **DeskPack plus**. Beide programma's maken evenals het vorige gebruik van het machinetaal programma **SYS POWER**. De ene listing is het conversieprogramma dat plaatjes uit PrintShop omzet naar standaard graphics. Er kunnen maximaal 9 plaatjes op het scherm worden gezet, waarbij u de keuze heeft over het aantal te tonen plaatjes. Het programma vraagt u, na het plaatsen van een PrintShop datadiskette, of het plaatje al dan niet op het scherm moet worden getoond. Als het aantal opgegeven plaatjes op het scherm staan kunt u ze als een file bewaren op diskette. U drukt hiertoe op de spatiebalk. Nu wordt een filenaam gevraagd. Voer een bestandsnaam in en druk op RETURN. Evenals de **icoon grabber** kunt u een weggeschreven scherm weer inladen en convertren naar een GeoPaint file.
(zie listing 'Converteren PrintShop')

Als u de plaatjes naar een GeoPaint file heeft geconverteerd kunt u ze bewerken in GeoPaint of bewaren in een zgn. Photo Album. Op deze manier kunt u een flinke bibliotheek met GEOS graphics opbouwen. De 3e listing, PrintMaster converter, is nagenoeg hetzelfde als de listing van het programma PrintShop converter. De werking ervan is gelijk als het converteerprogramma voor PrintShop plaatjes.
(zie listing 'Converteren PrintMaster')

Tot slot

Zoals u ziet zijn het nogal forse listings, maar ze zijn zeker de moeite van het intikken waard. De programma's werden geschreven door Patrick Cools. Op deze wijze kan er in Nederland en België een scala aan Public Domain software groeien. Dus, nogmaals een oproep aan de programmeurs onder u, stuur eens wat op. Leuke en originele GEOS programma's worden (mits gepubliceerd) gehonoreerd met een redelijke vergoeding. Ook serieuze toepassingen van de GEOS applicaties zijn van harte welkom. Volgende keer hopen we aandacht te besteden aan een assembler programma dat met behulp van GeoProgrammer moet worden ingevoerd. Mocht u niet in het bezit zijn van het programma **GeoPaint Catcher** dan kunt u de GEOS Public Domain disk 1 bestellen. U kunt dit doen door overmaking van *f* 12,00 op gironummer 1585491 t.n.v. Infolist Amsterdam o.v.v. GEOS INFO PD Disk. Rest mij nog u succes toe te wensen bij het intikken van de listings en veel genoegen bij het werken met de programma's.

Bert Venema

```
10 REM ******************************
20 REM *                            *
30 REM *CONVERTEREN VAN GEOS ICONEN*
40 REM *                            *
45 REM *  TOT STANDAARD GRAFIEKEN   *
50 REM *                            *
55 REM *    VOOR COMMODORE INFO     *
60 REM *                            *
70 REM *    DOOR PATRICK COOLS      *
80 REM *                            *
90 REM ******************************
100 IFSY=1THEN130
110 IFPEEK(828)=160THEN130
120 SY=1:LOAD"SYS POWER",8,1
130 DIMZ$(5),TH(50),SH(50),BY(72)
140 POKE53280,0:POKE53281,0:POKE646,1
150 SYS828:SYS862
160 PRINT"(CLR)(DOWN)(DOWN)     PLAATS GEOS DISK IN DE DRIVE"
170 PRINT"             (DOWN)(DOWN)>>EEN TOETS<<"
180 GETQ$:IFQ$=""THEN 180
190 S=1:I=1:S1=S
200 OPEN1,8,15:OPEN2,8,2,"#"
210 PRINT#1,"U1";2;0;18;S1
220 GET#2,T$,S$:S=ASC(S$+CHR$(0))
230 A=21
240 FORK=1TO8
250 PRINT#1,"U1";2;0;18;S1
260 PRINT#1,"B-P 2";A
270 GET#2,T$:TH(I)=ASC(T$+CHR$(0))
280 GET#2,S$:SH(I)=ASC(S$+CHR$(0))
290 A=A+32:I=I+1
300 NEXTK
310 S1=S
320 IFS<>255THEN210
330 GOSUB590
340 FORK=1TOI-1
350 IFTH(K)=0THEN430
360 PRINT#1,"U1";2;0;TH(K);SH(K)
370 PRINT#1,"B-P 2 5"
380 FORL=1TO63
390 GET#2,BY$:BY(L)=ASC(BY$+CHR$(0))
400 NEXTL
410 GOSUB680
420 IFA=31560THEN490
430 NEXTK
440 GOSUB640
450 PRINT"(DOWN)        (RVS)S(OFF)AVE OF (RVS)N(OFF)OG MEER ICONEN"
460 GETQ$:IFQ$<>"S"ANDQ$<>"N"THEN460
470 IFQ$="N"THENCLOSE2:CLOSE1:GOTO160
480 GOTO510
490 GOSUB640
500 PRINT"(DOWN)(DOWN)            SCHERM IS VOL"
510 PRINT"(DOWN)(DOWN)        (RED)GEOS DISK VERWIJDEREN(WHT)"
520 PRINT"(DOWN)(DOWN)          DATA DISK PLAATSEN"
530 CLOSE2:CLOSE1
540 INPUT"(DOWN)(DOWN)       FILENAAM";Z$(0)
550 POKE886,LEN(Z$(0))
560 SYS917
570 END
580 REM SUBROUTINE GRAFISCH SCHREM
590 POKE56576,PEEK(56576)AND254
600 POKE53265,PEEK(53265)OR32
610 POKE53272,64+32+16+8
620 RETURN
630 REM SUBROUTINE NORMAAL SCHERM
640 POKE53265,27:POKE53272.21
650 POKE56576,PEEK(56576)OR1
660 RETURN
670 REM SUBROUTINE SCHERM POKEN
680 READA:B=1
690 FORYY=0TO2
700 A1=A+(YY*320)
710 FORXX=0TO7
720 A2=A1+XX
730 FORII=0TO2
740 TT=II*8
750 POKEA2+TT,BY(B)
760 B=B+1
770 NEXTII
780 NEXTXX
790 NEXTYY
800 RETURN
810 DATA 24904,26184,27464,28744,30024,31304
820 DATA 24936,26216,27496,28776,30056,31336
830 DATA 24968,26248,27528,28808,30088,31368
840 DATA 25000,26280,27560,28840,30120,31400
850 DATA 25032,26312,27592,28872,30152,31432
860 DATA 25064,26344,27624,28904,30184,31464
870 DATA 25096,26376,27656,28936,30216,31496
880 DATA 25128,26408,27688,28968,30248,31528
890 DATA 25160,26440,27720,29000,30280,31560
```

```
20 REM *                            *
30 REM *   CONVERTEREN PRINTSHOP    *
40 REM *                            *
50 REM *   NAAR STANDAARD GRAFIEK   *
55 REM *                            *
60 REM *    VOOR COMMODORE INFO     *
65 REM *                            *
70 REM *     DOOR PATRICK COOLS     *
80 REM *                            *
90 REM ******************************
100 IF SY=1THEN130
110 IFPEEK(828)=160THEN130
120 SY=1:LOAD"SYS POWER",8,1
130 DIMZ$(9):SY=0
140 PRINT"(CLR)CONVERTEREN VAN PRINTSHOP PLAATJES"
150 POKE53280,0:POKE53281,0:POKE646,1
160 POKE52,92
170 PRINT"(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)TOT STANDAARD GRAFIEK"
180 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)MAX. 9 FILE'S"
190 INPUT"AANTAL FILE'S";AA
200 IF AA>9 THEN 190
210 PRINT"(DOWN)PLAATS P/S GRAPHICS DISK IN DE DRIVE"
220 GETQ$:IFQ$=""THEN220
230 OPEN 1,8,15,"I":CLOSE1
240 GOSUB590
250 POKE56576,PEEK(56576)AND254
260 POKE53265,PEEK(53265)OR32
270 POKE53272,64+32+16+8
280 SYS828:SYS862
290 LS=1
300 Z$(0)=Z$(LS):POKE886,LEN(Z$(0))
310 SYS902
320 B=32768:READA
330 FORY=0TO6
340 A1=A+(Y*320)
350 FORX=0TO7
360 A2=A1+X
370 FORI=0TO10
380 T=I*8
390 POKEA2+T,PEEK(B)
400 B=B+1
410 IFB=33340THEN450
420 NEXTI
430 NEXTX
440 NEXTY
450 LS=LS+1
460 IF LS >AA THEN 480
470 GOTO 300
480 GETX$:IFX$=""THEN480
490 POKE53265,27:POKE53272,21
500 POKE56576,PEEK(56576)OR1
510 PRINT"(DOWN)NU WORDT HET SCHERM GESAVED"
520 PRINT"(BLK)PLAATS EEN DATA DISK IN DE DRIVE(LBLU)"
530 PRINT"       DRUK EEN TOETS"
540 GETQ$:IFQ$=""THEN 540
550 INPUT"GEEF FILENAAM ";Z$(0)
560 POKE886,LEN(Z$(0))
570 SYS917
580 END
590 REM DIR LEZEN
600 I=0
610 OPEN1,8,0,"$0"
620 GET#1,A$,B$
630 GET#1,A$,B$
640 GET#1,A$,B$
650 GET#1,B$:IFST<>0THENCLOSE1:RETURN
660 IF B$<>CHR$(34)THEN650
670 GET#1,B$:IFB$<>CHR$(34)THEND$=D$+B$:GOTO670
680 GET#1,B$:IFB$<>CHR$(32)THEN680
690 C$=""
700 C$=C$+B$:GET#1,B$:IFB$<>""THEN700
710 IFI=0THENPRINTD$:I=I+1:GOTO760
720 PRINTD$;TAB(20)"(RVS)J(OFF)A/(RVS)N(OFF)EE"
730 GETQ$:IFQ$<>"J"ANDQ$<>"N"THEN730
740 IFQ$="J"THENPRINTTAB(20)"(UP)     JA":Z$(1)=D$:I=I+1:GOTO760
750 IFQ$="N"THENPRINT"(UP)                         (UP)":GOTO760
760 IFI>AATHENCLOSE1:RETURN
770 D$="":GOTO630
780 DATA24904,25008,25112
790 DATA27464,27568,27672
800 DATA30344,30448,30552
```

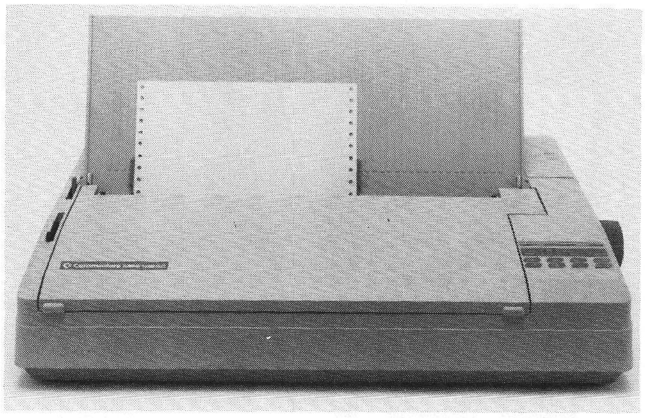
```
10 REM ****************************
20 REM *                          *
30 REM *  CONVERTEERT PRINTMASTER *
40 REM *                          *
50 REM *  TOT STANDAARD GRAPHICS  *
55 REM *                          *
60 REM *   VOOR COMMODORE INFO    *
65 REM *                          *
70 REM *    DOOR PATRICK COOLS    *
80 REM *                          *
90 REM ****************************
100 IF SY=1THEN130
110 IFPEEK(828)=160THEN130
120 SY=1:LOAD"SYS POWER",8,1
130 DIMZ$(9):SY=0
140 PRINT"(CLR)CONVERTEREN VAN PRINTMASTER GRAPHICS"
150 POKE53280,0:POKE53281,0:POKE646,1
160 POKE52,92
170 PRINT"(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)TOT STANDAARD GRAPHICS"
180 PRINT"(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)MAX. 9 FILE'S"
190 INPUT"AANTAL FILE'S";AA
200 IF AA>9 THEN 190
210 PRINT"(DOWN)PLAATS PRINTMASTER DATA DISK IN DE DRIVE"
220 GETQ$:IFQ$=""THEN220
230 OPEN 1,8,15,"I":CLOSE1
240 GOSUB600
250 POKE56576,PEEK(56576)AND254
260 POKE53265,PEEK(53265)OR32
270 POKE53272,64+32+16+8
280 SYS828:SYS862
290 LS=1
300 Z$(0)=Z$(LS):POKE886,LEN(Z$(0))
310 SYS902
320 B=32773:READA
330 FORY=0TO7
340 A1=A+(Y*320)
350 FORX=0TO7
360 A2=A1+X
370 FORI=0TO11
380 IFI=0THEN410
390 T=I*8
400 POKEA2+T,PEEK(B)
410 B=B+1
420 IFB=33398THEN460
430 NEXTI
440 NEXTX
450 NEXTY
460 LS=LS+1
470 IF LS >AA THEN 490
480 GOTO 300
490 GETX$:IFX$=""THEN490
500 POKE53265,27:POKE53272,21
510 POKE56576,PEEK(56576)OR1
520 PRINT"(DOWN)NU WORDT HET SCHERM GESAVED"
530 PRINT"(BLK)PLAATS EEN DATA DISK IN DE DRIVE(LBLU)"
540 PRINT"       DRUK EEN TOETS"
550 GETQ$:IFQ$=""THEN 550
560 INPUT"GEEF FILENAAM ";Z$(0)
570 POKE886,LEN(Z$(0))
580 SYS917
590 END
600 REM DIR LEZEN
610 I=0
620 OPEN1,8,0,"$0"
630 GET#1,A$,B$
640 GET#1,A$,B$
650 GET#1,A$,B$
660 GET#1,B$:IFST<>0THENCLOSE1:RETURN
670 IF B$<>CHR$(34)THEN660
680 GET#1,B$:IFB$<>CHR$(34)THEND$=D$+B$:GOTO680
690 GET#1,B$:IFB$<>CHR$(32)THEN690
700 C$=""
710 C$=C$+B$:GET#1,B$:IFB$<>""THEN710
720 IFI=0THENPRINTD$:I=I+1:GOTO770
730 PRINTD$;TAB(20)"(RVS)J(OFF)A/(RVS)N(OFF)EE"
740 GETQ$:IFQ$<>"J"ANDQ$<>"N"THEN740
750 IFQ$="J"THENPRINTTAB(20)"(UP)     JA":Z$(I)=D$:I=I+1:GOTO770
760 IFQ$="N"THENPRINT"(UP)                              (UP)":GOTO770
770 IFI>AATHENCLOSE1:RETURN
780 D$="":GOTO640
790 DATA24904,25008,25112
800 DATA27464,27568,27672
810 DATA30344,30448,30552
```

Het grootste deel van de post, die Bert Venema en ik in de GeosRubriek krijgen, bestaat uit brieven van mensen die de een of andere printer bezitten, maar die 'doet-ut-niet'. Dit artikel legt stap voor stap uit, hoe U zelf een printerdriver kunt schrijven. De gegeven voorbeelden zijn voor een 8 punts centronics-compatible printer.

Geos-machinetaalcursus (6)

Maak Uw eigen printerdriver

Printers zijn er in soorten en maten. Het GEOS-systeem, met een bijzonder sterke nadruk op grafische toepassingen, kan alleen met printers werken die ook grafische mogenlijkheden bezitten. Daisywheelprinters en Letterwielprinters vallen dus af. Op de mogenlijkheden van Laserprinters met GEOS (wat zeer wel te doen is, zie het pakket GeoLaser) zal ik, omdat dit natuurlijk weinig voorkomt, niet ingaan. Matrixprinters blijven dus over.



Matrixprinters hebben doorgaans twee modi: draft/gewoon/NLQ en de grafische modus. Een matrixprinter print altijd verticale rijen puntjes. In de eerstgenoemde modus haalt de printer de bit-patronen uit een in de printer-ROM opgeslagen karakterset. Wat de bit-patronen betreft is er geen verschil tussen de draft/gewoon/NLQ-print-stijlen, de verschillen worden bereikt door dubbel overprinten een en andere technieken. In de tweede modus (grafisch) worden de bit-patronen door de aansturende computer zelf geleverd.

Naalden

Een belangrijk kenmerk van een matrixprinter is het aantal naalden in de printkop. Tegenwoordig zijn er twee kategorieen voor printers: 9-naalds en 24-naalds. Als derde categorie zijn de zeer kromme 7-naalds printers te onderscheiden, bijvoorbeeld de Brother HR-5, en de MPS 801/803. (Mijn eigen printer valt natuurlijk in deze categorie.) Alle 7-naalds printers die ik ken, zijn echter MPS 801-compatible, en die werken feilloos (maar traaag) met de "Commodore" driver. Bezitters van deze printers zullen daarom o.h.a. geen aanstuurproblemen hebben. Dus terug naar de 9 en 24-naalds printers. Negen-naalds printers gebruiken eigenlijk maar 8 of 7 naalden in de grafische modus, de onderste naald(en) zijn bedoeld voor naar onder uitstekende karakters, zoals de "p" en de "g". In de handleiding valt te lezen hoeveel puntjes Uw printer verticaal print in de grafische (ook wel 'bit map') mode.

Serieel commodore

Sommige printers laten zich heel gemakkelijk op de C-64 aansluiten, dat zijn onder andere natuurlijk de commodore printers, maar ook modellen van andere fabrikanten, zoals de Star NL10C en de Okidata 120. (Dit zijn dus printers met een commodore seriële aansluiting, die direct kunnen worden aangesloten) Bijna alle andere printers die te koop zijn, hebben een centronics paralelle aansluiting. Met deze printers doen zich nu de meeste problemen voor.

Parallel centronics

In principe zijn er twee manieren om een parallelle centronicsprinter aan te sluiten op een C-64. Allereerst kan met een interface de centronics aansluiting worden omgezet in een seriële. Doordat er hiervoor nogal wat elektronica komt kijken, zijn deze dingen vrij duur. In illustratie 1 zijn alle door Berkeley Softworks goedgekeurde interfaces opgesomd. Er is echter nog een tweede manier die GEOS in het begin ook leek te ondersteunen. Het is namelijk ook mogelijk om een centronics-aansluiting met een simpel kabeltje direct op de user-poort van de C-64 aan te sluiten. De bekende Power Cartridge ondersteunt deze mogelijkheid bijvoorbeeld. De kabel die hiervoor nodig is, is zelf makkelijk te maken, en ook Berkeley Softworks had in 1986/1987 de zg. GeoCable in het GEOS-programma opgenomen. Echter, in 1988 heeft men, om mij onbekende redenen, nog voor er ook maar een printerdriver die met deze kabel samenwerkte op de markt was gekomen, besloten de Geocable te schrappen. De aansturing van printers aan de user-poort is een vrij ingewikkelde zaak, er moet dan gerommeld worden met protocols in de user-poort CIA2-adressen ($dd00 en verder). Deze keer zal ik dit dan ook niet behandelen, dat is misschien nog stof voor een volgend artikel. Mocht er iemand onder de lezers zijn die meer weet van het 'GeoCable-mysterie', dan zou ik daar graag meer over horen.

Listings

Wat ik nu dan wel ga behandelen is de methode om een werkende printerdriver te schrijven voor een printer die is aangesloten op de gewone seriële bus, direct of indirect (met een interface). Tussen de tekst van dit artikel door vindt u voorbeelden, die u, als u ze achter elkaar plakt de complete assembly-listing geven. Deze listing dient met de GeoAssembler gecompileerd te worden. Listing 1 bevat het programmaatje dat dan door de Geolinker gehaald moet worden. Als header kunt U de header van de vorige aflevering gebruiken, al moet U wel APPLICATION door PRINTER en InitRoutine (2x) door InitForPrint vervangen. De listing kan echter ook in een gewone assembler ingegeven, de code moet dan wel vanaf $7900 ingegeven worden, en naderhand met de Geos-Program-maker 2.0 (Listing 2) op disk gesaved worden. Let dan wel weer op, want in de listings komen macro's voor (bijv. LoadW en LoadB) en adres-labels (zoals r0,r3 etc). Deze zijn in de vorige afleveringen al uitgebreid uitgelegd, en dienen op de toen aangegeven wijze vertaald te worden. Tenslotte nog dit: de beschreven listing is slechts een voorbeeld en geldt niet voor iedere printer, want hoewel er eenzelfde stekker aan vast zit, lopen de printercommando's van de verschillende printers uiteen. De eigen waarden van de printercommando's zullen uit de handleiding gehaald moeten worden.

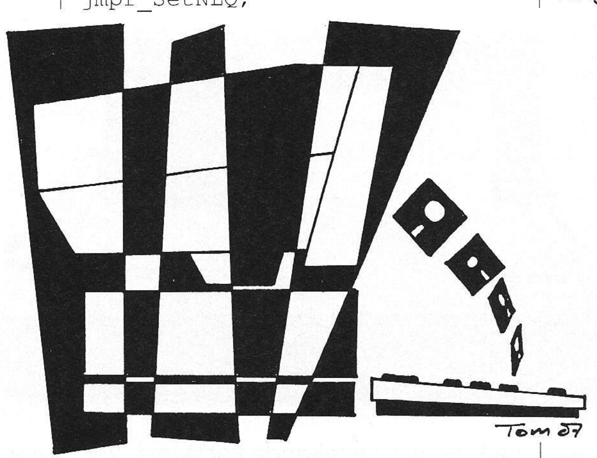
Printer Driver Definitie

Allereerst bekijken we wat algemene gegevens voor de printer. Het is voor het goede overzicht altijd beter deze in constanten te vangen:

```
devicenr= 4
;vul de eigen waarden in!
secadres= 5
;idem
breedte= 80
;in blokken van 8 punten breed
lengte= 90
;in blokken van 8 punten breed
LineFeed= 12
FormFeed= 15
CarRet= 13
```

Een GEOS-printerdriver begint altijd op adres $7900. De eerste 24 bytes zijn 8 jumps naar 8 routines. Deze jump-tabel is nodig, omdat op deze manier het adres van elke printerroutine gelijk gehouden kan worden voor alle verschillende printerdrivers. Zodoende kunnen de applicaties gewoon een vast adres voor iedere routine aanroepen, zonder te hoeven weten welke printerdriver in het geheugen zit.

```
.psect
.include GeosMac
.include GeosSym
InitForPrint:
jmpr_InitForPrint;
StartPrint:
jmpr_StartPrint;
PrintBuffer:
jmpr_PrintBuffer;
StopPrint:
jmpr_StopPrint;
GetDimensions:
jmpr_GetDimensions;
PrintASCII:
jmpr_PrintASCII;
StartASCII:
jmpr_StartASCII;
SetNLQ:
jmpr_SetNLQ;
```



Hierna begint de ruimte, waarin in de printer-data algemene informatie over de status van de printer wordt bijgehouden.

```
PrinterNaam:
.byte "Voorbeeldprinter",0
printbaarblok:
.byte 0,0,0,0,0,0,0,0
teller:.byte 0
hulppointer:
.byte 0,0
modeflag:.byte 0
```

De variabele modeflag dient voor gebruik in de printer, om te onderscheiden of er wel of niet in de grafische modus geprint wordt. (0=grafisch, 255=ASCII) Nu komen wat algemene routinetjes voor het openen en sluiten van kanalen.

```
OpenFile:
lda#devicenr
jsr$ffb1
lda#$f0+secadres
jsr$ff93
jsr$ffae
rts
CloseFile:
lda#devicenr
jsr$ffb1
lda#$e0+secadres
jsr$ff93
jsr$ffae
rts
OpenPrint:
lda#devicenr
jsr$ffb1
lda#$60+secadres
jsr$ff93
rts
ClosePrint:
jsr$ffae
rts
```

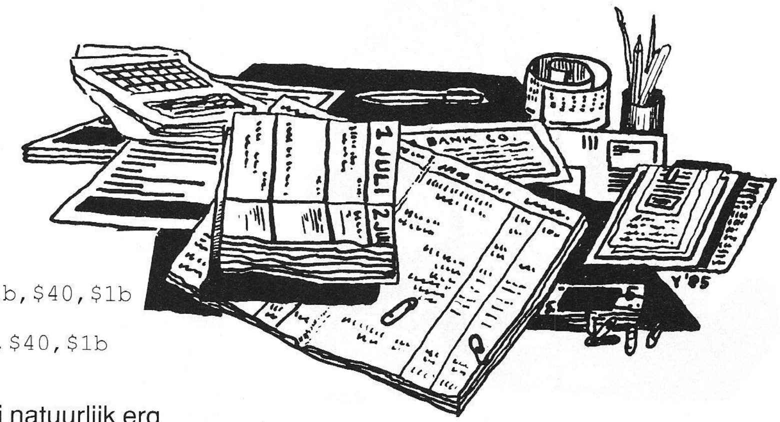
Twee afleveringen geleden werden alle routines van de printerdriver vanuit het oogpunt van de gebruiker (de applicatieprogrammeur) besproken: "wat kan je ermee?". Nu zal voor alle routines de implementatie (uitwerking) in machinetaal behandeld worden. Beginnen we bij het begin. De eerste routine, InitForPrint, wordt aangeroepen aan het begin van elk document (dus niet elke pagina!). Meestal doet deze procedure helemaal niets. De code hiervoor is dan ook frappant:

```
r_InitForPrint:
rts
```

Zo, dat is eruit. De volgende procedure, dat is Startprint, houdt wat meer in. Deze procedure wordt aangeroepen aan het begin van elke pagina, en dient de printer in de grafische modus te brengen, en de printfile te openen, dus wat in BASIC is: "open4,4 etc..".

```
r_StartPrint:
lda#0
stamodeflag
;0=graphics
GeneralStart:
lda#devicenr
jsrSetDevice
jsrInitForIO;
lda#0
sta$90
jsrOpenFile
lda$90
cmp#0
bnePrintError
jsrOpenPrint
jsrInitPrinter
jsrClosePrint
jsrSync
jsrDoneWithIO
ldx#0
rts
Sync:
```

```
ldy#0
1$:
ldx#0
2$:
inx
bne2$
iny
bne1$
rts
PrintError:
pha
jsrCloseFile
jsrDoneWithIO
pla
tax
rts
InitPrinter:
bitmodeflag
bmimode_ascii
;mode dus graphic
LoadW$08,s_InitGraph
lda#5
;lengte s_InitGraph
jmpPrintString
mode_ascii:
LoadW$08,s_InitAsc
lda#4
;lengte s_InitAsc
jmpPrintString
PrintString:
stateller
lus1:
decteller
ldyteller
lda($08),y
jsr$ffa8
cpy0
bnelus1
rts
s_InitGraph:
.byte
    $08,$41,$1b,$40,$1b
s_InitAsc:
.byte $32,$1b,$40,$1b
```



Omdat StartAscii natuurlijk erg lijkt op StartPrint, zal ik even uit de chronologie vallen en deze routine eerst bespreken.

```
r_StartASCII:
lda#$ff
stamodeflag
jmpGeneralStart
```

Zoals U ziet, de rest is gelijk aan StartPrint. In GeneralStart doet de computer steeds precies hetzelfde, tot InitPrinter. Hier wordt weer onderscheid gemaakt tussen de twee printmodi. Er moeten namelijk logischerwijze twee verschillende printer-commando's gegeven worden. Als de mode grafisch is, wordt s_InitGraph naar de printer gestuurd, anders (ascii-mode) s_InitAsc. Dus s_InitGraph is het commando om de printer in de grafische stand te brengen, s_InitAsc is het commando om de printer in de (gewone) ascii-karakterstand te brengen. De strings die geprint worden, hebben nog wel wat addertjes onder het gras. Ten eerste worden ze van achter naar voor geprint (door mijn PrintString-routine). Ten tweede is hier het punt gekomen waar U niet zomaar klakkeloos mijn listing kunt overnemen. U moet in Uw handleiding kijken welke karakters nodig zijn om Uw printer in de grafische stand te brengen! Deze "printer-commando's" zullen waarschijnlijk steeds uit twee karakters bestaan de ESC-code (27 ofwel $1b) en een andere waarde. Dit veelvuldig voorkomen van de waarde $1b is in mijn listing ook te zien. De door mij als voorbeeld gegeven printer-commando's werken overigens voor een Epson FX-80.

Omdat ik de volgorde nu toch al in de steek heb gelaten zal ik de routines nu in oplopende moeilijkheid doorlopen. Allereerst dus GetDimensions. Deze routine is bedoeld om de gebruiker (lees: de GEOS-applicatie-programmeur) te laten weten hoeveel puntjes de printer in de breedte aankan. Dit wordt aangegeven per blokje van 8. De meeste printer halen de standaard 640 puntjes (ofwel 80 blokjes) wel, maar onder andere de printer van Uw GEOS-redacteur komt maar tot 480 puntjes (60 blokjes). Deze routine doet dus helemaal niets, het zet alleen de breedte in x en de lengte in y, zodat de applicatie weet met wat voor een printer hij te maken heeft.

```
r_GetDimensions:
ldx#breedte
ldy#lengte
rts
```

Onder de verschillende matrixprinters zijn zeer uiteenlopende vormen van NLQ te vinden. De routine SetNLQ is bedoeld voor gebruik tijdens het ASCII printen. Hij opent gewoon een kanaal naar de printer en stuurt het NLQ-printer-commando door. Als Uw printer helemaal geen NLQ heeft, dan bestaat deze procedure gewoon uit een RTS. In de binnenkomende post zitten nogal eens mensen die hun printer tijdens de grafische mode, die bijna altijd door GEOS gebruikt wordt, NLQ willen laten doen. Als dit mogelijk is met Uw printer (zie Uw handleiding, of probeer het met een simpel BASIC-programmaatje eens), dan dient deze routine hier NIET voor. want deze routine was, zoals gezegd, bedoeld voor ASCII-printen. Als U dus NLQ in de grafische mode wilt, dan dient U het NLQ commando aan s_InitGraph (de string om de printer in de grafische mode te brengen) toe te voegen. U kunt ook -afhangende van de printer-veranderingen aanbrengen in s_GraphLine (komt nog) of de routine PrintBuffer iedere regel tweemaal laten afdrukken en dan pas een return geven. Maar nu de SetNLQ voor ASCII-printen:

```
r_SetNLQ:
lda#devicenr
jsrSetDevice
jsrInitForIO
  jsrOpenPrint
  LoadWr3,s_NLQon
  ;s_NLQon bevat de NLQ-string
  lda#4
  ;lengte van s_NLQon
  jsrPrintString
  jsrCloseprint
  jsrDoneWithIO
  rts
  s_NLQon:
  .byte $47,$1b,$45,$1b
```

StopPrint wordt aangeroepen aan het einde van elke pagina. Deze procedure print eventueel nog overgebleven data in de outputbuffer uit (lees verder), en produceert een formfeed (de printer draait de pagina eruit). Bovendien wordt de printfile, die in de routine StartPrint (of StartAscii) was geopend, nu gesloten.

```
r_StopPrint:
lda#devicenr
jsrSetDevice
jsrInitForIO
jsrOpenPrint
lda#FormFeed
jsrffa8
jsrClosePrint
jsrCloseFile
jsrDoneWithIO
rts
```

Zoals U zich mischien nog uit aflevering 4 kunt herinneren, reserveert de applicatie altijd 2 buffers van 640 bytes voor het printen. De ene buffer heet de applicatiebuffer, de andere de printer-driver buffer. Naar deze twee buffers wordt gewezen door respectievelijk r0 en r1. PrintAscii is een routine die een met een 0 beëindigde AS-

CII-string print in het standaardkarakter van de printer. Alleen de eerste buffer (applicatiebuffer) wordt gebruikt.

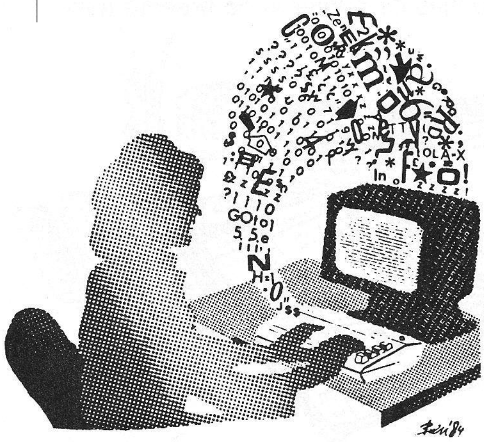
```
r_PrintASCII:
lda#devicenr
jsrSetDevice
jsrInitForIO
jsrOpenPrint
ldar0L
stahulppointer
ldar0H
stahulppointer+1
10$:
lda$hulppointer
cmp#0
beq40$
cmp#CarRet
bne20$
jsr$ffa8
lda#LineFeed
20$:
jsr$ffa8
clc
inchulppointer
bcc30$
inchulppointer+1
30$:
jmp10$
40$:
jsrClosePrint
jsrDoneWithIO
rts
```

De laatste routine is PrintBuffer, deze print een grafische regel van 640 bytes uit. De 640 bytes zijn op de bekende C-64 manier gerangschikt, dus de eerste 8 bytes onder elkaar, daarnaast de volgende 8 bytes: dus 80 blokjes van 8 bytes naast elkaar.

```
r_PrintBuffer:
lda#devicenr
jsrSetDevice
jsrInitForIO
jsrOpenPrint
ldar0l
star3l
ldar0h
star3h
jsrPrintPrintBuffer
lda#Carret
jsr$ffa8
lda#LineFeed
jsr$ffa8
jsrClosePrint
jsrDoneWithIO
rts
```

7-naalds trouble

Het ruige werk begint nu. Het uitprinten van een buffer van 640 bytes in de grafische mode is ingewikkeld, omdat, zoals ergens hierboven vermeld is, printers in verticale rijen bits printen. Ergens zullen dus de bytes uit de buffer op hun kant gezet moeten worden (roteren), en dat in machinetaal! Helemaal feestelijk wordt het, als er met een 7-naaldse printer gewerkt wordt; de aangeleverde buffer is immers 8 puntjes diep, dus blijf je steeds met een rij zitten! Wegens plaatsgebrek kan ik geen volledige listing geven van de wijze waarop een 7-naaldse driver moet werken. Het algemene idee is echter dat na 7 maal printen er 7 rijen extra over zijn, zodat er dan twee regels geprint kunnen worden. Het precieze algorithme (de werkwijze) staat beschreven in illustratie 2. Er is hier overigens nog een lastig puntje: mocht de tekening niet uit een door 7 deelbaar aantal verticale blokken bestaan en het programma is klaar met printen, dan is er nog een aantal rijen die nog niet geprint zijn, opgeslagen in de eerste buffer. Daarom moet de routine StopPrint (eerder besproken) voor 7-naalds printers ook deze resterende rijen uitprinten.



```
PrintPrintBuffer:
jsrTestBuffer
;kijkt of er iets te printen
   is
;in teller het aantal blokken
ldateller
bne5$
;niets, dan gauw klaar
rts
5$:
jsrSetGraphics
;geeft printcommando's die
   nog gedaan
;moeten worden voordat er 640
   bytes
;naar de printer gedumpt
   kunnen worden.
lda#breedte
sec
sbcteller
tax
10$:
txa
pha
jsrRotate
;pakt een blok van 8x8,
   roteert dit
;en zet het in printbaarblok.
jsrSendBuff
;zendt inhoud van
   printbaarblok naar
;printer
clc
lda#8
adcr3L
lda#0
adcr3H
pla
tax
dex
bne10$
;volgende blokje
rts
;SUBROUTINES:
TestBuffer:
clc
lda#[((breedte-1)*8)
;lowbyte!
adcr3l
lda#]((breedte-1)*8)
;highbyte!
adcr3h
r3 nu op begin laatste blokje
lda#breedte
stateller
5$:
ldy#0
  10$:
  lda(r3),y
  cmp#0
  bne15$
  iny
  cpy#8
  bne10$
  clc
  lda#8
  sbcr3l
  bcc$20
  decr3h
  20$:
  decteller
  bne 5$
  15$:
  rts
  SendBuff:
ldx#0
10$:
txa
pha
ldaprintbaarblok,x
jsr$ffa8
pla
tax
inx
cpx#8
bne10$
rts
Rotate:
sei
ldy#7
10$:
(r3),y
ldx#7
20$:
rora
rorprintbaarblok,x
dex
bpl20$
dey
bpl10$
cli
rts
```

Als U een 9-naalds printer heeft met een 8-punts grafische modus, of een 24-naaldse met een 8-puntse grafische modus, dan kunt U de bovenstaande routines bijna wel letterlijk overnemen. De enige moeilijkheden ontstaan bij de procedure SetGraphics. Deze procedure dient de commando's naar de printer te zenden die per regel gegeven moeten worden, rechtstreeks voordat de 640 bytes grafische data verzonden worden.

```
SetGraphics:
LoadBr3h,#0
ldateller
star3l
aslr3l
rolr3h
aslr3l
rolr3h
aslr3l
rolr3h
sec
lda#[(breedte*8)
;'['=lowbyte van een word
sbcr3l
stas_GraphLine+1
lda#](breedte*8)
;']'=highbyte van een word
sbcr3h
stas_GraphLine
LoadWr3,#s_GraphLine
lda#5
;lengte s_GraphLine
jmpPrintString
s_GraphLine:
.byte 0,0,4,"*",ESC
EndAdress:
```

Zoals U gezien hebt, bestaat een printer-driver alleen maar uit een uitwerking in machinetaal van de routines, die 2 afleveringen geleden bij U geïntroduceerd werden. Vele printer-drivers zullen daarom op elkaar lijken, en het zal vaak voorkomen, dat de commandostrings (s_.....) het enige verschilpunt zijn. De hier geLISTe printer-driver lijkt bijvoorbeeld sprekend op de Epson FX- 80 driver (standaard in het GEOS-pakket) en op de driver die in de GEOS-Reference Guide staat. Als U een goede (of verbeterde) driver wilt schrijven aan de hand van dit artikel, dan zijn er twee methodes, volgens welke U kunt werken:

1) Reset de computer op een moment, wanneer U er zeker van bent, dat de printer-driver die U wilt veranderen in de computer zit (bijvoorbeeld onder het printen met GeoWrite). De driver is dan zoals altijd in het geheugengebied $7900 en verder te vinden. U kunt de driver dan met een machinetaalmonitor aanpassen, en dan met Listing2 op disk zetten. Het nadeel van deze methode is, dat de machinetaallisting die U van de monitor krijgt, natuurlijk geen labels heeft. Er zal dan ook flink gezocht moeten worden naar de onderdeeltjes die veranderd dienen te worden.

2) Gewoon opnieuw beginnen, en de driver van de grond af opbouwen. Dit is wat meer werk, maar het is misschien wel zo verstandig. Hoe dan ook, ik hoop dat dit artikel uw printerproblemen heeft kunnen oplossen. Volgende keer misschien meer over printers aan de user-poort.

Peter Boncz

O wat kan het leven toch mooi zijn! Het warme zonnetje door de ramen, een vers kopje thee en vooral veel computers om je heen om te experimenteren. Glory, glory halleluja it's just me, myself and my computer! We gaan weer in voor een volgende ronde, daar is ie weer!!

Invullers en wegdragers

Weer een splinternieuwe aflevering van de tips & trucs hoek. We hebben weer een bescheiden selectie gemaakt uit onze voorraad. Langzaam aan beginnen nu ook de tips en trucs van de lezers binnen te komen. Maar het is nog niet genoeg, WE NEED MORE STUFF! Dus, opsturen naar het wellicht al veel te vaak genoemde maar nog niet bekend genoeg zijnde adres....

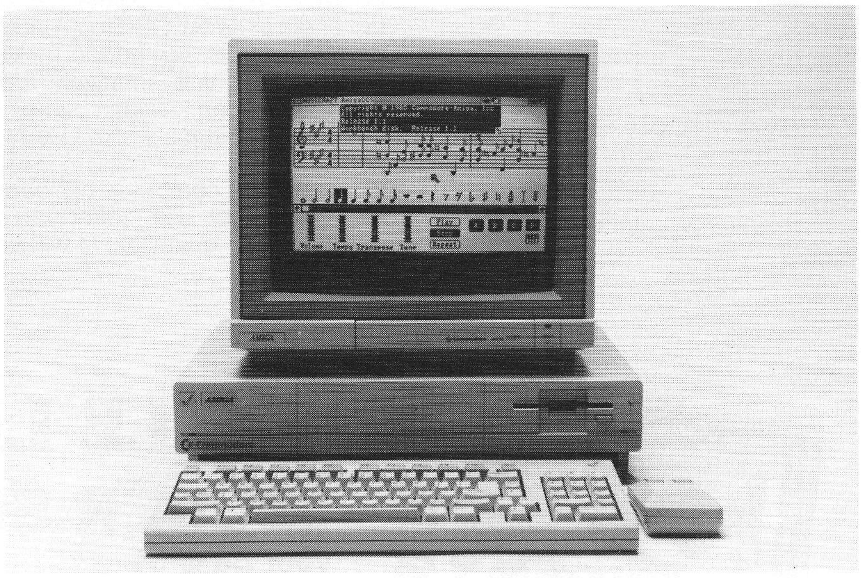
Binnenin AmigaDOS

De eerste lag al weer enige tijd op antwoord te wachten. Sander van der Kooij uit Bleiswijk schreef deze aan ons. Hij had een aantal vragen over de startup-sequence.

° 1. Als ik nu met het install commando van een disk een bootdisk maak, kan ik hier dan ook een Startup-Sequence op zetten (zodat ik de disk-icon kan openen van deze schijf)

° 2. Kun je in de startup-sequence ook het 'say'-commando gebruiken?

° 3. Kun je in de startup-sequence ook een 'graphic' op het scherm zetten (IFF afbeeldingen) door het te laden.



ad 1. Eerst het antwoord op de eerste vraag. Dit is inderdaad mogelijk, Sander. Even een korte inleiding. Je hebt de disk geformatteerd met 'format' en geïnstalleerd met 'install'. De disk zal nu in ieder geval als bootdisk gebruikt kunnen worden. Maak nu eerst een directory met de naam 's' aan. Voer vervolgens de onderstaande batchfile in met ED met de naam "s/startup-sequence" :

```
loadwb
endcli >nil:
```

Als je nu een aantal programma's met hun bijbehorende icon informatie op schijf hebt staan, dit zijn de files welke dezelfde naam als het programma hebben alleen ze hebben de '.info' eraan toegevoegd gekregen dan kun je ze vanuit de Workbench openen met de muis.

ad 2. Het is inderdaad mogelijk mbv. het 'say'-commando gesproken tekst, vanuit de startup-sequence, hoorbaar te krijgen. We gaan het nu alleen hebben over Workbench 1.3. De syntax van SAY luidt;

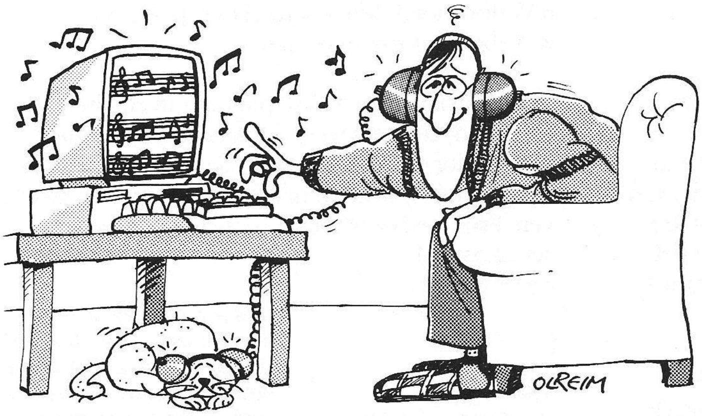
```
SAY [opties] "string"
```

Bij de opties valt te kiezen uit *-m, -f, -r, -n, -s###, -p###, -x.*
Deze opties staan achtereenvolgens voor, male, female, robot intonatie, natuurlijke intonatie, speed (40-400), pitch (65-320) en de inhoud van achter -x opgegeven filenaam hoorbaar maken.
3. Standaard is het niet mogelijk om een grafische afbeelding op het scherm te zetten. Hiervoor zijn enkele PD utilities te krijgen. Elders in dit blad zul je ook een routine vinden, in C, waarmee je een IFF plaatje kunt laden. Dit kan ook vanuit de startup-sequence gebeuren.

Commodore 1901 en Amiga?

De nu volgende reactie kwam van de heer Hanekamp uit Delfzijl. Dhr. Hanekamp is in het bezit van een Commodore 128 met 1901 monitor. Hij wil nu een Amiga 500 aanschaffen. Nu vroeg hij of het mogelijk was deze monitor ook op de Amiga aan te sluiten.
Geachte meneer Hanekamp. We hebben voor u wat na zitten vragen. Het blijkt met veel geknutsel wel mogelijk te zijn, echter het is niet aan te raden. De Amiga 500 beschikt over enkele video-uitgangen. Ze heeft een composiet uitgang, een RGB TTL uitgang en een RGB lineair uitgang. De composiet-uitgang kan zonder meer op elke composiet ingang aangesloten worden. Denk hierbij aan de composiet ingang van uw 1901, een videorecorder ingang en de ingang van een monochroom monitor. Hierbij krijgt u alleen maar een beeld met verschillende grijstinten. De volgen-

de, de RGB TTL-uitgang. Deze is in wezen gelijk aan de uitgang die op uw Commodore 128 zit. Met deze uitgang is het maximale aantal kleuren echter beperkt tot 16, immers er is een Rood-, een Groen-, een Blauw- en een Intensiteit- lijn. Er zijn dus 4 verschillende kleurcombinaties mogelijk, dus dit zijn 2^4 = 16 kleuren. Niet bepaald 'the Amiga way'. Dus blijft alleen de RGB lineair uitgang over. De 1901-monitor beschikt niet over een RGB lineair-ingang dus dit wordt moeilijk. Er is nog een tussenoplossing welke geen aanpassing aan uw 1901-monitor vereist. Koop een TV-adaptor. Deze kosten plusminus *f* 90,=. produceren, naast een signaal voor een TV-toestel ook een composiet kleurensignaal. Deze kunt u wel weer op uw 1901 aansluiten. Toch levert dit niet het beeld dat de 1084(S) of soortgelijke zal opleveren. Dan is er nog een laatste radicale oplossing. Bouw uw 1901 om. Er zijn enkele methodes bekend. Een Duits blad heeft recentelijk een schema gepubliceerd om een 1901 monitor geschikt te maken voor RGB lineair. Toch raden we u aan, als u niet thuis bent in de TV-technieken, dit niet zelf te doen.



Niets lukt!

W. Willemse uit Winterswijk schreef ons een brief waarin hij duidelijk maakte dat wat hij ook probeerde, hem niets lukte. Geen programma kwam goed, de voorbeelden wilden bij hem niet werken, etcetera.
Het is nog steeds zo, dat de BASIC-programma's van de Amiga zonder checksummer ingetypt moeten worden. We zijn druk doende een goed lopende checksummer in elkaar te knutselen. Tot die tijd zult u moeten proberen de listings foutloos in te typen. Eén goede fout en het programma gaat goed onderuit.
Waarschijnlijk zullen uw fouten, meneer Willemse, ook wel uit dezelfde bron voortkomen. Het blijft dus een zaak de programma's te 'debuggen' tot het bittere einde!

Sample problemen

Het blijft nog steeds een verrassende zaak, de geluidsmogelijkheden van de Amiga. De Amiga registreert geluid door het analoge geluid om te zetten naar een digitaal equivalent. Deze slaat zij op in haar geheugen. Als de gebruiker er om vraagt zal zij het geluid weer omzetten naar haar analoge tegenhanger om dat vervolgens naar een aan de Amiga gekoppelde versterker te sturen. En voilà, daar klinken de muzikale hoogstandjes u om de oren. Echter, één nadeel. Een mooi geluid vereist veel geheugen. Veel geheugen in de computer en veel geheugen op de harddisk, cq. diskdrive. J.B. Dekker uit Apeldoorn kwam met het volgende probleem te zitten. Door de lengte van de geluidsamples duurt het onaanvaardbaar lang om ze weer in te lezen in BASIC. Bestaat hiervoor een mogelijkheid om ze, bv. met behulp van een library, sneller in te lezen?
Ja, mijnheer Dekker, hiervoor bestaat inderdaad wel een oplossing. We zijn eens in onze oude programma's gaan zitten snuffelen en hebben hieruit een oplossing geëxtraheerd (mooi woord he!...uit de Van Dalen gehaald!...). Bekijk het onderstaande programma en experimenteer ermee. Nog even uw hint voor extra tips & trucs, we zitten er altijd verlegen om dus stuur ze maar op!!

```
 DECLARE FUNCTION xOpen&
LIBRARY
 DECLARE FUNCTION xRead&
LIBRARY
 DECLARE FUNCTION Al-
locMem& LIBRARY

 LIBRARY "dos.library"
 LIBRARY "exec.library"

 INPUT "Filenaam :";naam$
 INPUT "Hoeveel bytes inle-
zen :";bytes%

 'Met xOpen is het moge-
lijk een bestand te openen
 'De syntax van xOpen :
pointer& = xO-
pen&(naam+chr$(0),mode&)
 'de modes zijn : MODE_RE-
ADWRITE = 1004 (lezen en
schrijven)
 '
MODE_OLD           = 1005 (al-
leen lezen)
 '
MODE_NEW           = 1006 (al-
leen schrijven)

 fp&=xO-
pen&(SADD(naam$+CHR$(0)),100
5)

 IF fp&=0 THEN PRINT "File
bestaat niet!":END

 MemOpts&=2^0+2^16
 BufSize&=bytes%+5
 BufAdr&=AllocMem&(BufSi-
ze&,MemOpts&)

 'Met xRead leest men gege-
vens van de file naar een
door de
 'gebruiker aangegeven
buffer. Deze dient men aan
te maken met
 'de exec-functie AllocMem
 'De syntax van xRead :
 'aantal_gelezen = xRe-
ad&(pointer&,buffera-
dres&,aantal_te_lezen%)

 aantal = xRead&(fp&,Bu-
fAdr&,bytes%)

 IF aantal  bytes% THEN
PRINT "Uitgelezen :";aantal
 PRINT aantal;" bytes inge-
lezen vanaf adres
$";HEX$(BufAdr&)

 PRINT "Ik sluit de file
nu weer af en geeft het ge-
heugen weer vrij!"

 'Met xClose sluit men de
met xOpen geopende file
weer af
 'De syntax : xOpen&(poin-
ter&)

 CALL xClose&(fp&)
 CALL FreeMem&(BufAdr&,Buf-
Size&)
```

Tips en Trucs Wanted!!
Stuur ze op naar Commodore Info, Postbus 43048, 1009 ZA, Amsterdam, en vermeldt er bij 'Amiga Tips & Trucs'

Amiga en haar karakter(s)

Normale karakters op het scherm weergeven is bij de Amiga 'een fluitje van een cent'. Toch hebben de meeste Amiga-gebruikers nog grote moeite met de bijzondere karakters, zoals de trema's, ringel-s'en, etcetera. Uiteraard kun je via SetMap een Frans of een Duits toetsenbord kiezen maar dat geeft weer andere problemen. Er is echter een veel eenvoudigere oplossing. Een woord als'schön' tikt u als volgt in:

° tik eerst normaal 'sch'

° druk op de LINKER ALT-toets en houdt deze ingedrukt

° druk nu op de lettertoets 'K' (er gebeurt niets zichtbaars)

° laat de ALT-toets los en tik een 'o'

en zie: achter de 'sch' verschijnt een 'ö'. Nu nog een 'n' er achter en 'schön' staat er bepaald 'schön'!
Ditzelfde kunt u doen voor alle andere accenten. Gebruik hiervoor de LINKER ALT-toets in combinatie met de lettertoetsen F, G, H en J.
Rechtstreeks kunt u bijvoorbeeld intypen: de plusminus, een combinatie van de LINKER ALT-toets en de 'z'.
Gaat u alle mogelijkheden maar eens na en probeer ook de combinatie SHIFT-, LINKER ALT- en alle letter- en cijfertoetsen. Deze methode werkt altijd: in de CLI, in Amiga-BASIC etcetera.

J.B. Dekker, Apeldoorn.



Hexa- en decimaal

Het blijft altijd weer lastig, het omzetten van decimaal naar hexadecimaal en omgekeerd in een C-programma. We hebben hiervoor een kleine oplossing. Voeg de onderstaande listing in uw programma in.

```
char *dtoh(getal)
int getal;
{
 char hex[2];
 int diglo,dighi;

 dighi=getal/16;
 diglo=getal-(16*dighi);

 if(dighi<0x0a)
   hex[0]=dighi+0x30;
 else
   hex[0]=dighi+0x37;

 if(diglo<0x0a)
   hex[1]=diglo+0x30;
 else
   hex[1]=diglo+0x37;

 return(hex);
}

htod(getal)
char *getal;
{
int dec;

if(getal[0]<0x3a)
dec=((int)toupper(getal[0])-0x3
    0)*16;
else
dec=((int)toupper(getal[0])-0x3
    7)*16);

if(getal[1]<0x3a)
dec=dec+(int)toupper(getal[1])-
    0x30;
else
dec=dec+(int)toupper(getal[1])-
    0x37;

return(dec);
}
```

De aanroep van beide routines verschilt een klein beetje. De routine **htod()** kan gewoon als een functie aangeroepen worden, dus:

```
dec_getal = htod("hex_getal");
```

De routine **dtoh()** verschilt een klein beetje. Hierbij moet de uit de routine gekomen string gekopieerd worden in een andere string. Een voorbeeld:

```
main()
{
 char hex_getal[2];

 strcpy(hex_getal,dtoh(10));
}
```

Hierbij zal dan '0a' gekopieerd worden in de string 'hex_getal'.

Vreemd geprinte tekens!?

Vreemde tekens op het scherm krijgen is mogelijk, zie het eerste rubriekje. Hoe krijgen we deze karakters nu op de printer? Er is wel wat voorbereidend werk nodig, maar daarna is het gewoon weer een kwestie van de juiste karakters via het toetsenbord invoeren. In listing 1 ziet u een invoersubroutine. Ook al zou u de printer niet willen gebruiken, deze subroutine heeft toch wel een aantal voordelen, te weten:

° ALLE karakters kunnen worden ingevoerd, dus ook de in het eerste rubriekje genoemde tekens, komma's, puntkomma's en zelfs aanhalingstekens.

° de cursor is op het scherm te zien als een professioneel ogend streepje.

° u kunt het aantal in te voeren karakters beperken (iets wat met INPUT of LINE INPUT niet mogelijk is).

Nu weer terug naar de printer. U kunt zien dat in listing 1 de plaats is aangegeven waar de printerinstructies moeten komen. Listing 2 is een voorbeeld hoe deze instructies zouden kunnen luiden. Houdt er echter wel rekening mee dat in dit voorbeeld rekening is gehouden met gebruik van een Commodore printer, echter in de meeste gevallen zal deze listing correct zijn. Mocht het toch nodig zijn de codes aan te passen, raadpleeg dan de handleiding van uw printer. Wijzig dan de codes aan de RECHTERZIJDE van de listing. Mocht u eventueel niet weten wat de CHR$ aan de LINKERZIJDE beteken, dus welke codes het zijn, typ ze in het BASIC uitvoervenster in en het teken wordt afgeprint.

J.B. Dekker, Apeldoorn.

Listing 1a. Hoofdprogramma

```
CLS: PRINT
    TAB(25)"INVOER-SUBROUTINE"
FOR x=1 TO 5
PRINT: CALL Invoer
a$(x)=tekst$
NEXT: CLS
FOR x=1 TO 5:PRINT a$(x): NEXT
END
```

Listing 1b. Invoersubroutine

```
SUB Invoer STATIC
SHARED tekst$
tekst$="":z$=""
invoerlus:
PRINT CHR$(95);CHR$(8);
z$=INKEY$
IF z$="" THEN invoerlus
IF z$=CHR$(8) THEN GOSUB
    herstel:GOTO invoerlus
IF z$=CHR$(13) THEN terug
ac=ASC(z$)
IF ac>26 AND ac<32 THEN
    invoerlus
verder:
PRINT z$
```

```
printercodes:
'=========================
'hier moeten de printer
'codes komen
'=========================
nogverder:
tekst$=tekst$+z$
lengte=LEN(tekst$)
IF lengte>78 THEN klaar
GOTO invoerlus
terug:
IF tekst$=" " THEN
    tekst$=CHR$(13)
tekst$=LEFT$(tekst$,lengte-1)
PRINT z$
RETURN
klaar:
END SUB
```

Listing 2. Printercodes

```
IF z$=CHR$(196) THEN
    z$=CHR$(142)
IF z$=CHR$(228) THEN
    z$=CHR$(132)
IF z$=CHR$(203) THEN
    z$=CHR$(144)
IF z$=CHR$(235) THEN
    z$=CHR$(137)
IF z$=CHR$(214) THEN
    z$=CHR$(153)
IF z$=CHR$(246) THEN
    z$=CHR$(148)
IF z$=CHR$(220) THEN
    z$=CHR$(154)
IF z$=CHR$(252) THEN
    z$=CHR$(129)
IF z$=CHR$(232) THEN
    z$=CHR$(138)
IF z$=CHR$(233) THEN
    z$=CHR$(130)
IF z$=CHR$(239) THEN
    z$=CHR$(139)
IF z$=CHR$(231) THEN
    z$=CHR$(135)
IF z$=CHR$(223) THEN
    z$=CHR$(225)
IF z$=CHR$(187) THEN
    z$=CHR$(175)
IF z$=CHR$(171) THEN
    z$=CHR$(174)
IF z$=CHR$(188) THEN
    z$=CHR$(172)
IF z$=CHR$(189) THEN
    z$=CHR$(171)
IF z$=CHR$(177) THEN
    z$=CHR$(241)
```

rnd in C

Nee, dit is geen begin van een klassiek meesterstuk. Ook is het geen kinderrijmpje. Gewoon simpel, hoe krijg ik random getallen in C?
Het probleem was, waar krijg ik de 'seed' waarden vandaan. 'Seed' waarden zijn de basisgetallen waar aan de hand van het randomgetal wordt berekend. Na enig gezoek leek het ons niet al te moeilijk. De CIA's van de Amiga bevatten beide een EventCounter. Dit is een heel snel lopende teller. Pak het onderste byte, deze 'loopt' immers het snelst en tralaa... daar is je 'Seedvalue'. De routine:

```
float rnd(x)
int x;
{
UBYTE *Seed=(UBYTE *)0xbfd800;
return(float*Seed/255.00);
}
```

Onthoud wel, de include file 'exec/types.h' moet meegecompiled worden, immmers we gebruiken de variabele UBYTE. Deze is geen standaard C variabele en dient dus gedefinieerd te worden. Nog één opmerking, onthoudt dat de teruggekregen waarde van het type 'float' is. Men dient dus ook de floating point library mee te linken!



Copperaardigheden

Grafische grapjes doen het nog steeds goed in computerland vandaar deze. Het programmaatje zorgt voor 3 kleurige balken op het beeldscherm in de kleuren blauw, rood en groen. Het aardige van dit demootje is dat de truc met de Copper wordt gedaan. Het programmeren van de Copper gebeurt niet vanuit machinetaal, C of Modula 2, nee gewoon vanuit AmigaBasic!!
De Copper is één van de co-processoren welke zich tot taak gerekend heeft om op bepaalde tijdstippen registers van de Custom chips met bepaalde waarden te vullen.
Het programmeren van de Copper gebeurt met drie commando's die de Copper begrijpt, namelijk Wait, Move en Skip. Alleen de eerste twee worden regelmatig gebruikt. Het commando Wait zorgt ervoor dat de Copper wacht tot een bepaalde electronenstraal positie wordt bereikt en met Move kunt u een register een bepaalde waarde geven. Op die manier kan een programma gemaakt worden die de Copper stuurt. Een dergelijk programma noemt men een Copperlijst. Hoe kunnen we nu een Copperlijst vanuit BASIC programmeren. Bekijk het volgende programma eens en experimenteer er eens wat mee:

```
DECLARE FUNCTION ViewAddress&
    LIBRARY
DECLARE FUNCTION
    ViewPortAddress LIBRARY
DECLARE FUNCTION AllocMem&
    LIBRARY

LIBRARY "intuition.library"
LIBRARY "graphics.library"
LIBRARY "exec.library"

COLOR 1,0
PALETTE 0,0,0,0
Initialize
WaitC 0,20
MoveC 384,0
MoveC 386,4095

'Fade van zwart naar blauw en
    tekst van wit naar blauw

FOR l%=100 TO 115
   WaitC l%,20
   MoveC 384,l%-100
   MoveC
    386,4095-(256*(l%-100))-(16
    *(l%-100))
NEXT l%

'Fade van blauw naar rood,
    tekst blijft gelijk

FOR l%=116 TO 131
   WaitC l%,20
   MoveC
    384,131-l%+(256*(l%-116))
NEXT l%

'Fade van rood naar groen,
    tekst blijft gelijk

FOR l%=132 TO 147
   WaitC l%,20
   MoveC
    384,3840-(256*(l%-132))+(16
    *(l%-132))
NEXT l%

'Fade van groen naar zwart,
    tekst van blauw naar wit

FOR l%=148 TO 162
   WaitC l%,20
   MoveC 384,240-((l%-148)*16)
   MoveC
    386,15+(256*(l%-148))+(16*(
    l%-148))
NEXT l%

Activate
LIBRARY CLOSE
END

SUB Initialize STATIC
  SHARED UCopList&
  MemOpts&=2^0+2^1+2^16

    UCopList&=AllocMem(12,MemOp
    ts&)
  IF UCopList&=0 THEN ERROR 7
END SUB
```

```
SUB Activate STATIC
   SHARED UCopList&
   WaitC 10000,256

     viewport&=ViewPortAddress&(
     WINDOW(7))
   POKEL viewport&+20,UCopList&
   CALL RethinkDisplay
END SUB

SUB WaitC(y%,x%) STATIC
   SHARED UCopList&
   CALL CWait(UCopList&,y%,x%)
   CALL CBump(UCopList&)
END SUB

SUB MoveC(reg%,waarde%) STATIC
    SHARED UCopList&
    CALL
     CMove(UCopList&,reg%,waarde
     %)
    CALL CBump(UCopList&)
END SUB
```

Beeldige grapjes

De nu volgende routines zijn niet zozeer als onafhankelijk programma nuttig, het zijn echte tips. Wat u nodig heeft, kunt u eruit halen. Wat doen ze? Het zijn, zoals de titel al verraadt, routines die grapjes met het beeldscherm uithalen. De eerste is een programma in BASIC. Het zorgt ervoor dat het beeldscherm met de muispointer mee verschuift. Hoe wordt dit gedaan, zult u zich afvragen. Op zichzelf niet zo moeilijk. In de ViewPort van het scherm worden de x- en y-offset van het scherm opgeslagen. Door deze waarden aan te passen en vervolgens de routine 'RemakeDisplay()' aan te roepen zal het beeldscherm op een andere plaats te zien zijn. De tweede routine is rond dezelfde basis opgebouwd. Het is een routine in C welke ervoor zorgt dat het beeldscherm als een gek begint te trillen.

- Routine 1.

```
DECLARE FUNCTION ViewAddress&
    LIBRARY
LIBRARY "intuition.library"

ViewPort&=ViewAddress&
WHILE MOUSE(0)=0
dx%=66+MOUSE(1)/5.8
dy%=14+MOUSE(2)/2.5
IF dx%>128 THEN dx%=128
POKEW ViewPort&+12,dy%
POKEW ViewPort&+14,dx%
CALL RemakeDisplay
WEND

LIBRARY CLOSE
END
```

Zoals u ziet, geen ingewikkelde routines of berekeningen. De tweede routine is in C. Deze is ook al niet veel moeilijker.

- Routine 2.

```
#include "functions.h"
#include "exec/types.h"
#include "graphics/view.h"
#include
    "intuition/intuitionbase.h"


struct IntuitionBase
    *IntuitionBase;

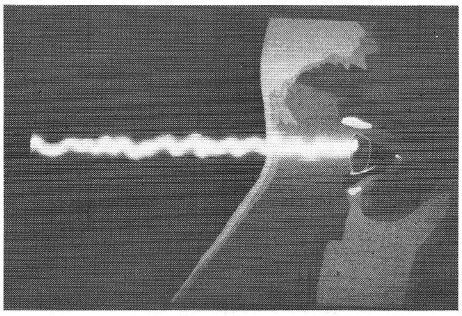
main()
{
 int i,t;
 char *mouse=(char *)0xbfe001;

 if(!(IntuitionBase=(struct
    IntuitionBase *)

    OpenLibrary("intuition.libr
    ary",0L)))
    {
     print("Geen intuition\n");
     exit(FALSE);
    }
 while((*mouse&0x40)==0x40)
    {
     for(i=20;i<40;i++)
      {

    IntuitionBase->ActiveScreen
    ->ViewPort.DyOffset=(SHORT)
    i;

    IntuitionBase->ActiveScreen
    ->ViewPort.DxOffset=(SHORT)
    i;
        RemakeDisplay();
        for(t=0;t<5000;t++);
      }
    }
 CloseLibrary(IntuitionBase);
}
```



TestDrive

Herinnert u het zich nog? De eerste dag dat u testdrive in uw bezit had. Als een kind zo blij. Tegen de tikkende klok racen, nou ja, tegen de bergwand racen zul je bedoelen, of het ravijn IN racen. Goed, er is een oplossing om toch met 140-160 mph. door de bochten te racen zonder dat u in de slip raakt en dat er ongelukken gebeuren. Houdt tijdens de bocht de vuurknop ingedrukt totdat u weer op het rechte stuk zit. Tataaa! Geen problemen. En nu maar hopen dat er zometeen de tekst verschijnt: 'Did you see any low flying planes'.

Muis Cleanex

Heeft u er ook al last van gehad? Vooral als u uw Amiga al wat langere tijd bezit? De muis reageert niet meer zo snel als dat hij vroeger deed. Of is dat maar verbeelding. Nee, dat is geen verbeelding. De muis wordt mechanisch gestuurd, dat wil zeggen, de bol in de muis draait twee rolletjes rond welke uiteindelijk via electronische uitleesmethoden voor de x en y pulsjes zorgen. Deze rolletjes kunnen op een gegeven moment zo baden in het stof dat ze niet meer accuraat kunnen reageren. Vooral bij snelle bewegingen maken ze er helemaal een puinhoop van. Wat er aan te doen. Schroef de muis open (is de garantie al verlopen?!?) en haal het stof bij de rolletjes vandaan. U kunt ze niet missen. Ze zitten in het gat waar de bal hoort te rollen.
Wat een smerige boel hè, al dat stof!

Diskdrives zonder schakelaar?

Beschikt u over zo'n losse diskdrive zonder schakelaar? Er zijn programma's, vooral als niet over 512K uitbreiding beschikt, die verlangen dat u de diskdrive ontkoppelt. Dit betekent dat u de stekker van de drive uit de Amiga moet halen. Dit is lastig en op de lange duur zelfs funest voor uw stekkers. Dan maar zelf een schakelaar er op bouwen. Het is niet zo lastig. Pak uw Amiga handboek erbij. Kijk naar de aansluiting genaamd 'external disk connector'. Op lijn 1 staat het RDY signaal. Door deze lijn door te knippen en er een schakelaar ertussen te zetten bereikt u het gewenste.

PokeMania!

Er zijn nogal wat manieren om je Amiga goed onderuit te laten gaan. Hieronder zijn een aantal poke's verzameld waardoor bepaalde functie Amiga veranderd kunnen worden. We staan niet helemaal in voor de gevolgen, want de kans is niet klein dat u uw Amiga opnieuw op moet starten. Gebruik AmigaBasic voor het invoeren van de poke's.

```
poke 8394,43          : de
    cursor wordt uitgeschakeld
poke 498736982,243 : de drive
    motor wordt aangezet
    (hopelijk geen
    read-errors!)
poke 498736982,0      : de
    drive motor gaat weer uit.
poke 564748398,54  : het
    audio-filter en de LED
    gaat uit
poke 564748398,12  : het
    audio-filter en de power
    LED gaan weer aan.
poke 12132,98         : een
    andere manier om je
    diskdrive naar de
    filistijnen te helpen.
    Haal de diskette eruit en
    waaauw! Schijnbaar
    overleeft de drive het
    wel.
poke 12137,x          : voor x
    een getal van 1 tot 15.
    Dit bepaalt de snelheid
    waarmee de diskdrive
    gecontroleerd wordt op
    een nieuwe schijf. (x=45
    geen controle)
poke 1214,234         : een
    verandering in
    AmigaBasic. Pak de
    startschijf alvast maar
    weer. Typ in; 'Load "een
    programmanaam"' en geef
    RETURN. Zo, die zit goed
    vast nietwaar!
poke 8388,76 en poke 8394,12
    zijn eigenlijk de moeite
    niet waard want je
    computer zit
    onherroepelijk vast.
poke 8967,34          : geen
    muispointer meer totdat
    de muis weer bewogen
    wordt.
poke 8977,34          :
    helemaal geen muispointer
    meer, ook niet als je de
    muis beweegt.
poke 8977,12          : aaah,
    daar is de muispointer
    weer!
```

We besluiten met de woorden van de persoon die dit alles instuurde, we hopen dat jullie Amiga nog leeft na deze poke's. Om de boel af te sluiten volgt een beeldflitser.

```
REM Beeldflitser
FOR a=0 TO 255
   POKE 30000000,a
NEXT a
RUN
```

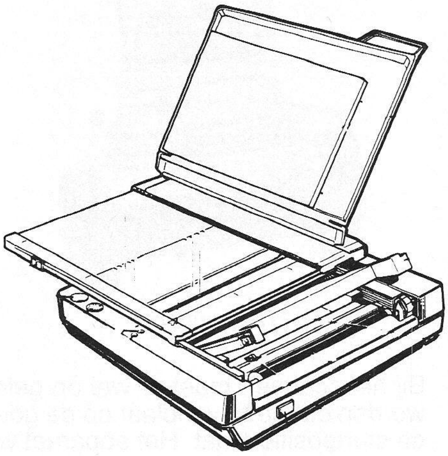
Tot Slot

Dit was het weer, luitjes! Ons koekblik is voor deze keer weer leeg, de postvakken zijn omgekeerd en onze hersenen staan weer op zes uur! Voordat we ophouden eerst nog even onze sponsor:
"De oproep om meer 'response' geldt nog steeds want we hebben nimmer genoeg informatie. We hebben uw hulp nodig om deze hoek zijn levensvatbaarheid te laten behouden. Daarom insturen die tips en trucs, zodat we u van een constante stroom van informatie kunnen voorzien."
Leuk hè, zo'n jingle tussendoor. Het lijkt wel een radioprogramma! Goed, morgenstond brengt goud in de mond zegt men, wij houden het hier voor gezien.

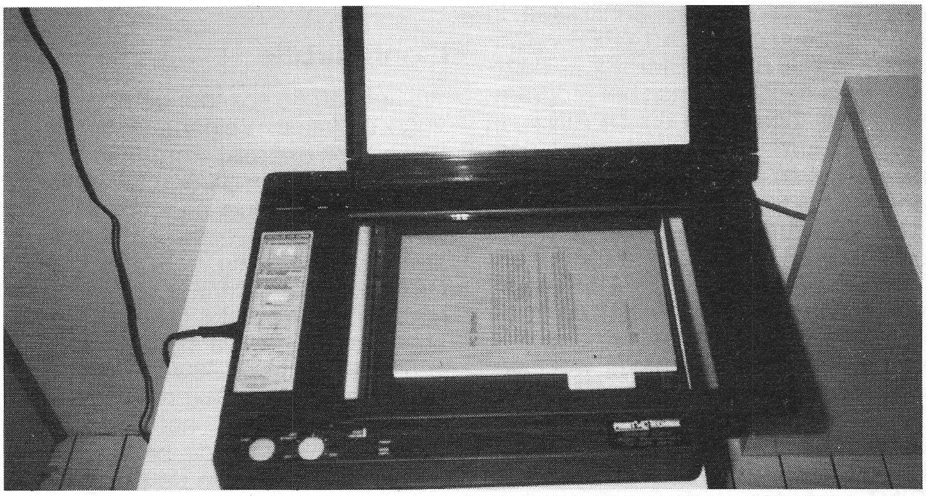
Scannen, kopiëren en printen met één apparaat. Dat is het gemak ten top. En als de kwaliteit van al die afdrukken dan ook nog eens van goede kwaliteit is, dan heb je voor *f* 1400,- wel iets moois in huis.

Universal scanner

Scannen, kopiëren en printen in één



Het scannen en daarna op de computer bewerken van illustraties, foto's, dia's en andere plaatjes is een veel toegepaste techniek onder computerkunstenaars, reclamemakers en uitgevers. Vaak wordt voor dit soort bewerkingen echter zeer dure apparatuur gebruikt. Ook de Amiga-hobbyist kan, met de Universal scanner en software als De Luxe Paint, Digipaint of Photonpaint de artistieke aspiraties achter het beeldscherm botvieren. Het resultaat kan daarna via dezelfde scanner uitgeprint worden. Andere toepassingen waar het apparaat geschikt voor is zijn desktop publishing, het aanleggen van een graphics database, zakelijk gebruik e.d.



Compact apparaat

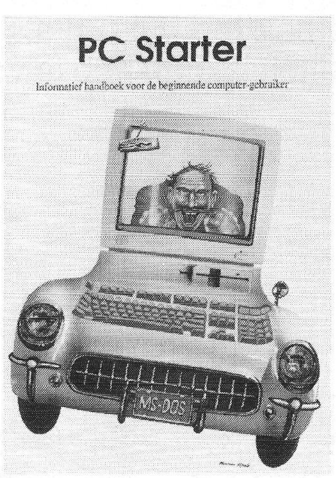
De Universal scanner is een slank uitgevoerd apparaat van 40 bij 36 en 11 centimeter hoog. Een roze en lichtblauwe bedienknop aan de voorkant geven een extra tintje aan het stemmige grijs van de machine. De roze knop start de kopieerprocedure en met de blauwe knop kan de weergave van de kopie of de scan traploos van licht naar donker ingesteld worden. Twee LEDs geven aan of het apparaat aan staat en of de papierrol op is. Aan de linkerkant zitten de aan/uit knop en het stroomcontact.
Het vervangen van de - thermische - papierrol is een fluitje van een cent. Je drukt op de knop aan de rechterkant van het apparaat, er floept een deksel omhoog en de rol kan er ingestopt of uitgelicht worden. In het deksel zit de thermokop gemonteerd.
De bovenplaat van de scanner beweegt bij het kopiëren van rechts naar links over de vaste fluorescentielamp en de plaat kan ook nog eens een stukje omhoog, zodat ook dikke boeken op de glasplaat gelegd kunnen worden. Aan het eind van de rit gaat het deksel niet vanzelf terug naar de beginpositie, maar moet met de hand teruggeschoven worden.
De verbinding met de computer zit eenvoudig in elkaar. Een kabel wordt aan de parallelle poort van de Amiga verbonden en de centronics connector aan de scanner.

Scannen

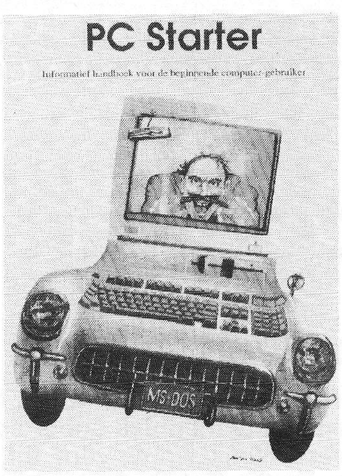
De scanner geeft alleen zwart-wit beelden. Als de software diskette in de drive wordt gestopt en de Amiga aangezet, volgt het booten en verschijnt de workbench waarvandaan via een icoon het scanproces gestart kan worden. Omdat de workbench ook geheugen opslokt kan men, als er maar 1 MB geheugen aanwezig is, niet het maximaal aantal van 2048 lijnen scannen. Om dit te voorkomen moet men tijdens het opstarten Ctrl-D ingedrukt houden en vanuit de CLI 'scanner' intikken.
Vanaf het scanner titelscherm kunnen twee resoluties aangegeven worden. Je hebt de keus uit een 640 x 512 of 640 x 256 resolutie. Bij de 640 x 512 mode kun je kiezen of de menu items in hun eigen venster blijven en geactiveerd worden door de muis, of dat de items permanent aan de rechterkant van het scherm zichtbaar blijven. Met de menu items wordt de scanner gestuurd. Items zijn Start Scan, 1:1, 1:2 en 1:4, FlipV, FlipH
1:1 geeft de beste resolutie in twee grijstinten. Van het origineel is 1/16 deel op het scherm zichtbaar en dit deel kan over het scherm bewogen worden.
1:2 geeft de scan in vijf grijstinten weer en 1/4 van het plaatje is zichtbaar. Ook dit deel kun je over het scherm verplaatsen.
1:4 geeft het beeld in volledig weer in 16 grijstinten.
Met de Flip-opties kan het beeld gekanteld worden.

Menu

Behalve de bovengenoemde items is er aan de bovenkant een balk met pull-down menu's; het project-menu voor bestandsoperaties; het scanmenu voor het veranderen van de scan parameters zoals aantal scanlijnen en contrast; het screen menu voor saven en printen en het window menu voor het markeren en bewaren van delen van het scanoppervlak.



Het origineel



Gescand en daarna geprint

Bij het scannen moet er wel op gelet worden dat de bovenplaat op de goede startpositie staat. Het apparaat wil de volledige afstand afleggen. Als de plaat dus niet bij het begin staat en aan het eind komt, stopt de scan. Als de bovenplaat vervolgens een stukje terug geduwd wordt gaat het scannen verder. Er verschijnt in ieder geval geen mededeling op het beeldscherm als de plaat niet vanaf de startpositie begint.

Bewerken en printen

Als de scan bewerkt moet worden moet men het eerst via een van de save-functies wegschrijven. Daarna kan het in diverse tekenpakketten ingelezen en bewerkt worden.
Om een IFF-bestand op de printer af te drukken wordt in het screen menu de print optie gekozen. Alleen datgene wat op het scherm staat wordt afgedrukt. Grijstonen worden omgezet in een zwart-wit raster.
Met de huidige software worden 440 pixels horizontaal afgedrukt.

Printen vanuit DeLuxe Paint

Het printen vanuit bijvoorbeeld DeLuxe Paint III gaat als volgt: met de scannerdiskette opstarten en met Ctrl-D onderbreken. Intikken Assign Thermo: RAM: (voor 2 MB), DH0: (voor hard disk), DF0: of DF1: (voor diskdrive). Daarna LOADWB en vanuit de Workbench op het icoon CMD gaan staan en één keer aanklikken. In het Infomenu de TOOLTYPES veranderen in THERMO:TEMP.FILE. Daarna vanuit de Workbench het programma CMD opstarten en DPAINT. In DPAINT kun je nu het plaatje laden, bewerken en printen. Hij gaat nu niet printen, maar schrijft het bestand weg naar de eerder opgegeven drive of disk, onder de naam TEMP.FILE. Vervolgens verlaat je DPAINT en bij minder dan één MB is het beter de computer te resetten wegens te weinig geheugen. In dit geval weer onderbreken met Ctrl-D en ASSIGN THERMO: naar de eerder gekozen optie (disk of drive). Vanuit de Workbench het programma THERMOPRINT opstarten en deze zal vanuit de geassignde drive het programm TEMP.FILE uitprinten. Het is dus een nogal omslachtige manier van doen, maar dat heeft alles te maken met het geheugen van 1 MB. Het resetten dient immers om het geheugen leeg te maken. Als er meer geheugen aanwezig is, dan wordt de procedure sterk ingekort.

Kwaliteit

Volgens opgave van de fabrikant bedraagt het oplossend vermogen van de scanner 200 dpi en worden de originelen in 16 grijstinten weergegeven. De afdrukkwaliteit is aanmerkelijk beter dan wat de gemiddelde matrixprinter voor elkaar krijgt, maar een laserprinter, met meestal een oplossend vermogen van 300 dpi, drukt diepere zwarten af en geeft een hoger contrast. Maar daar betaal je dan ook voor.
Ook moet je deze universele machine niet vergelijken met een volwassen kopieerapparaat, dat meer contrast kan weergeven, maar al met al is de kwaliteit van de scans en de afdrukken zeer bevredigend.
Opvallend is de snelheid waarmee de machine werkt. Binnen tien seconden rolt de afdruk van een A4-pagina eruit en het scannen gaat even snel.

Configuratie

Wij hebben de scanner op een Amiga 2000 uitgetest. Volgens de fabrikant kan hij ook gebruikt worden in combinatie met de Amiga 500 of de 1000. Wel is een geheugen van minimaal 1 MB nodig en liever meer. Verder is een diskdrive nodig.
De Universal scanner is gemaakt door het Duitse Print Technik. Op het apparaat, onder de bewegende bovenplaat, zijn gebruiksaanwijzingen in de Duitse taal geplakt maar dat mag geen bezwaar zijn: het apparaat is zeer eenvoudig in de bediening. Bij het apparaat worden de nodige software en parallelle kabel geleverd, evenals een rol thermopapier en een Engelstalige handleiding. Ook zit er een stift die als een gummetje voor het thermische papier werkt.
Van de Universal scanner bestaat ook een Atari- en IBM-versie. Hij wordt in Nederland geleverd door Cat & Korsh tegen de prijs van *f* 1399,-

Inl: Cat & Korsh, 010 - 4507696
J.B./H.S.



Een full colour strip werd met de Universal Scanner gekopieerd. De zwart/wit afstelling werkt nauwkeurig.

De meeste Amiga-bezitters beschikken wel over het tekenpakket D'Paint. Hoe schrijft dit pakket zijn tekeningen weg naar schijf. Of meer algemeen: is er een standaard manier voor het bewaren van graphics- en muziekdata, etcetera.

I.F.F-bestanden

Wedden dat veel Amiga-bezitters zeggen, 'tja, daar heb ik wel eens van gehoord. Doe je dat in je koffie of zo?'. Nee dus. IFF staat voor Interchange File Format wat zich vrij laat vertalen met 'Vrij Uitwisselbaar File Formaat'. Het is een, door Electronic Arts ontwikkeld, file-formaat. Binnen dit file-formaat kunnen verschillende soorten programma's hun gegevens opslaan. Zo'n programma kan een tekenprogramma zijn, maar het mag ook een tekstverwerkingspakket of sample-programma zijn. Om er nu voor te zorgen dat programma's ook weer te weten kunnen komen om wat voor soort gegevens het gaat is, gebruik gemaakt van een aantal vaste identifiers, een soort keywords, op vaste posities in de file.

De structuur

Hoe zit een IFF-file nu in elkaar (zie ook tabel 1)? Om te beginnen moet het programma, dat de file uit zal lezen, weten dat het hier om een IFF file gaat. Een IFF file zal daarom ALTIJD beginnen met de letters 'FORM'. Hierna volgt een longword waarin de lengte van de file vermeld is. De eerste 8 bytes, FORM + de lengte, is het zogenaamde header blok.
In de volgende 4 bytes wordt aangegeven om wat voor soort file het gaat, immers we praten hier over een gestandaardiseerd formaat. Een programma dat de file uitleest, moet er dan ook weer achter kunnen komen om wat voor IFF-file het gaat. Er zijn er meerdere. We noemen er hier 4, **WORD, ILBM, SMUS en 8SVX**.

Tabel 1. Schematisch een IFF file.

Offset	Omschrijving
0	FORM
4	de lengte van de gehele file in bytes
8	het IFF type
12	chunknaam
16	lengte van de chunk in bytes
20	de eigenlijke 'chunk'
...etcetera...	

WORD is de indentifier voor een tekstverwerkingspakket. ILBM geeft aan dat de IFF-file een tekening is. SMUS wordt door bepaalde muziekcomponeer (lichtelijk onhandig uitgedrukt!) programma's gehanteerd. Tenslotte is er nog 8SVX. Dit wordt in de IFF-file gezet door sommige geluidsample programmatuur.



Na het IFF-type volgen, afhankelijk van het IFF-type, enkele 'chunks'. Dit zijn gegevensblokken waarin data vermeld zijn over specifieke eigenschappen van de file. Even een voorbeeld. Stel het gaat om een ILBM file. Hierin worden grafische gegevens weggeschreven (data van D'Paint o.a). Het programma dat de gegevens uitleest moet weten hoe hoog en breed het scherm is. Ze moet ook weten hoeveel kleuren en welke kleuren er gebruikt moeten worden. Dit soort gegevens worden opgeslagen in de zogenaamde 'chunks' (zie tabel 2). De lengte van een 'chunk' is variabel, immers de 'chunk' waarin de beeldschermdata staan zal niet even lang zijn als de 'chunk' waarin de kleurgegevens opgeslagen zijn. Het longword volgend op de chunknaam, bevat de lengte van de 'chunk'. Eerst maar even een voorbeeld. Afbeelding 1 omvat een deel van ILBM-file zoals gegenereerd wordt door D'Paint. Het eerste longword bevat de letters 'FORM'. Hierop volgend komt de lengte van de totale file. In het voorbeeld bedraagt de lengte $62CE, hetgeen decimaal 25294 is. Het IFF-type is ILBM. Nu volgen de verschillende 'chunks'. (Zie tabel 3) De eerste is BMHD. De lengte is $14. Dus $14 bytes (=20 decimaal) verder volgt de volgende 'chunk'. Dit is CMAP. De CMAP 'chunk' is $30 bytes lang. Daarna volgen nog enkele chunks genaamd CRNG. Dezen hebben voor ons nu even geen betekenis. Als laatste is er dan een van de belangrijkste chunks, de BODY-chunk. Hierin zijn de eigenlijke beeldschermdata opgeslagen.

Tabel 2. De 'chunks'

Type	Chunk	Omschrijving
WORD	BODY	De gegevens van een afbeelding
	COLR	kleur van de tekst
	DOC	soort tekst
	FOOT	voetregel
	TABS	tabafstanden
	PINF	informatie over de afbeeldingen
	PARA	paginalayout
	FSCC	informatie over de tekstkleur
	HEAD	kopregels
	PCTS	welke afbeeldingen in de tekst
SMUS	SHDR	eigenschappen van het geluid (tempo, geluid-sterkte, etc)
	NAME	naam van het muziekstuk
	(c)	copyright vermelding
	AUTH	auteursnaam
	ANNO	aantekening bij het muziekstuk
	TRAK	definitie van het geluidskanaal
ILBM	BMHD	grafische eigenschappen
	CMAP	colortable
	BODY	het eigenlijke 'plaatje'
8SVX	VHDR	eigenschappen (type, tempo, octaaf, sterkte)
	NAME	naam van het geluid
	(c)	copyright
	AUTH	auteursnaam
	ANNO	aantekening bij het geluid
	BODY	het 'eigenlijke' geluid
	ATAK	de 'attack'
	RLSE	de 'release'

De duik in het diepe

We hebben al een voorbeeldfile bekeken. Dit was er slechts maar één. Hoe zien andere IFF files eruit. Hiervoor hebben we een kleine BASIC-routine geschreven. Typ deze in, en laat deze op verschillende IFF-files los. Het programmaatje spreekt hopelijk voor zichzelf. Het is maar een klein demonstratie programmaatje om te laten zien hoe in BASIC IFF-files gelezen kunnen worden.

Waar het allemaal om ging

Het uitgangspunt van dit artikel was een reactie van een lezer. Deze vroeg of er CLI-routines bestonden om een IFF-plaatje bij het opstarten op het beeldscherm te krijgen. Op de Workbench schijf zult u géén commando aantreffen die zoiets doet, dus werd het tijd voor ons om onze luie hersenen weer eens aan het werk te zetten. We gaan weer terug naar de tekentafel. We willen een ILBM-file inlezen. Hoe ziet een ILBM-file eruit? Als eerste chunk is er de BMHD. Dit betekent BitMap HeaDer. Hierin staan de volgende gegevens:

Offset	Type	Omschrijving
0	Word	breedte
2	Word	hoogte
4	Wordx	offset
6	Wordy	offset
8	Byte	diepte
9	Byte	masker
10	Byte	compressie (0=geen)
11	Byte	-
12	Word	transparant color
14	Bytex	aspect
15	Bytey	aspect
16	Word	breedte van de bronpagina
18	Word	hoogte van de bronpagina

De tweede 'chunk' die altijd in een ILBM file wordt aangetroffen is CMAP. Zoals we al eerder zagen worden hierin de ColorMap gegevens opgeslagen. Het gaat dus om de kleurgegevens van de afbeelding. Hoe ziet CMAP er nu uit?

Offset,	Type & Omschrijving
0	Byterood, component van kleur 1
1	Bytegroen, component van kleur 1
2	Byteblauw, component van kleur 1
3	Byterood, component van kleur 2
n	Byteblauw, component van kleur n

Als laatste belangrijke, de BODY 'chunk'. In deze chunk staan, zoals we al hebben gezegd, de eigenlijke beeldscherm data.

Er zijn nog enkele andere 'chunks' zoals CAMG, CCRT, etc. Wilt u hier en over andere zaken van ILBM files meer weten dan raden wij u aan hierover een goed boek aan te schaffen. Denk hierbij aan 'het supergrafiekboek voor de Amiga' van Data Becker.

Nu we alle gegevens hebben is het mogelijk een routine te schrijven waarme we het plaatje kunnen laden. Een probleem hierbij is dat BASIC eigenlijk te traag wordt om dit karwei snel en efficient uit te voeren. We hebben daarom onze toevlucht tot C gezocht.

Het programma

Typ het programma in en compile en link het met de Aztek C compiler. Zorg ervoor dat tijdens het linken het voorgecompilede programma jiff_save.o in de actuele directory staat. De routine jiff_save.c staat op disk SYS2 in de directory 'jiff'. Deze dient u al in gecompilede vorm, dus de '.o'-file, in de actuele directory te zetten.

```
0000: 464F524D  000062CE  494C424D  424D4844FORM..b.ILBMBMHD
0010: 00000014  02800100  00000000  04020100 ...............
0020: 00000A0B  028000C8  434D4150  00000030  ........CMAP...O
0030: 000000F0  D0B0E0E0  D0F00000  F0000000  ...............
0040: 9090E000  00A06060  00909060  50506050  ......"..'PP'P
0050: 60C0B0B0  0000F000  A0607070  70806060  '........'ppp."
0060: 43524E47  00000008  24780500  00010405  CRNG....$x......
0070: 43524E47  00000008  080E0280  00010405  CRNG...........
0080: 43524E47  00000008  2B9E0060  00010405  CRNG....+..'....
0090: 43524E47  00000008  395E0024  00010405  CRNG....9^.$....
00A0: 424F4459  0000622E  B100B100  B100B100  BODY..b.........
...enz...
```

Afbeelding 1. Een ILBM file

Als laatste

We hopen hiermee weer een aantal mensen geholpen te hebben. U kunt de instructie 'Loadilbm' in uw c-directory zetten. Het is hiermee mogelijk uw met D'Paint getekende plaatjes bij het opstarten van de Amiga te laten afbeelden. Heeft u meer suggesties of vragen omtrent dit of andere onderwerp(en), schroom dan niet te schrijven!

Johan & johan

Programma 1. Een IFF lezer

```
------------------------------
INPUT "Hoe heet de file";naam$

OPEN naam$ FOR INPUT AS 1

form$=INPUT$(4,1)
IF form$<>"FORM" THEN PRINT
    "Dit is geen IFF file":
    END

FrmLen=0
FOR l=0 TO 3
   temp$=INPUT$(1,1)

   FrmLen=FrmLen+ASC(temp$)*(2
   56^(3-1))
NEXT l

PRINT naam$;" is een IFF file"
PRINT
    "------------------------"
PRINT "De lengte         =
    ";FrmLen

type$=INPUT$(4,1)
PRINT "Het IFF Type     =
    ";type
teller=4
BTeller=0
WHILE (teller =< (FrmLen-4))
    BTeller=BTeller+1
    temp$=INPUT$(4,1)
    teller=teller+4
    PRINT "------------------"
    PRINT "Chunknaam          =
    ";temp$

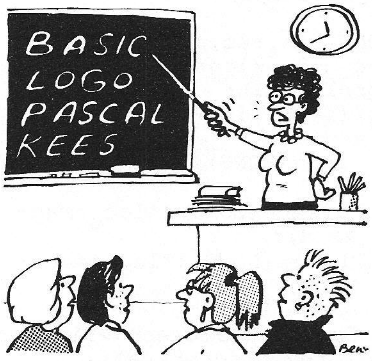
    ChkLen=0
    FOR l=0 TO 3
      temp$=INPUT$(1,1)

   ChkLen=ChkLen+ASC(temp$)*(2
   56^(3-1))
    NEXT l
    PRINT "Chunklengte        =
    ";ChkLen
    teller=teller+4+ChkLen
    IF ChkLen < 256 THEN
      temp$=INPUT$(ChkLen,1)
    ELSE
      Blocks=INT(ChkLen/256)
      Rest=ChkLen-(256*Blocks)
      FOR l=0 TO Blocks
         temp$=INPUT$(255,1)
      NEXT l
      temp$=INPUT$(Rest,1)
    END IF
    IF BTeller > 5 THEN
      WHILE a$=""
        a$=INKEY$
      WEND
     BTeller=0
     END IF
WEND

CLOSE #1
```

Instructies bij Loadilbm

Programmanaam : **Loadllbm**
Programmeertaal : **C**
Gebruikt programma : **Aztek C v3.6a**
Opties :
cc loadilbm.c +L
cc jiff_save.c +L
ln loadilbm.o jiff_save.o -lc32
Uitvoer : **Loadllbm filenaam**



Programma 2. Loadilbm.c

```
------------------------------
#include "stdio.h"
#include "functions.h"
#include "exec/types.h"
#include
    "intuition/intuitionbase.h"

#include "graphics/gfxbase.h"

#define IDVAL(a,b,c,d)
    ((LONGa <<24L | (LONG)b
    <<16L | c << 0|d)
#define ID_FORM
    IDVAL('F','O','R','M')
#define ID_ILBM
    IDVAL('I','L','B','M')
#define ID_BMHD
    IDVAL('B','M','H','D')
#define ID_CMAP
    IDVAL('C','M','A','P')
#define ID_BODY
    IDVAL('B','O','D','Y')

struct IntuitionBase
    *IntuitionBase=0;
struct GfxBase *GfxBase=0;
struct Screen *PScreen=0;
struct NewScreen Scherm;
struct IlbmBmhd = {
 UWORD Width,Height;
 UWORD x,y;
 UBYTE Depth;
 UBYTE Masking;
 UBYTE Compression;
 UBYTE Unused;
 UWORD Transparant;
 UBYTE X_Aspect,Y_Aspect;
 UWORD SrceWidth,SrceHeight;
} MyBmhd;

UBYTE
    *loc_bmhd,*loc_cmap,kleuren
    [100];
ULONG colors;

Quit(arg)
char *arg;
{
 printf("%s",arg);
 if(PScreen != NULL)
    CloseScreen(PScreen);
 if(GfxBase != NULL)
    CloseLibrary(GfxBase);
 if(IntuitionBase != NULL)
    CloseLibrary(IntuitionBase)
    ;
 exit(20);
}

BYTE GetByte(fp)
FILE *fp;
{
 BYTE read;

 read=getc(fp);
 if(feof(fp))
   Quit("EOF error\n");
 if(ferror(fp))
   Quit("File error\n");
 return(read);
}

ULONG GetLong(fp)
FILE *fp;
{
 ULONG read;
 read=0L;

    read=(ULONG)IDVAL(GetByte(f
    p),GetByte(fp),GetByte(fp),
    GetByte(fp));
 return(read);
}

LoadPicture(name)
char *name;
{
 FILE *fp;
 LONG id;
 void LookForPic();

    if((fp=fopen(name,"r"))==NU
    LL)
     Quit("Opgegeven plaatje
     is niet op schijf!\n");
 id=GetLong(fp);
 if(id!=ID_FORM)
     Quit("Geen IFF file,
    FORM error\n");
 LookForPic(fp);
 fclose(fp);
}

OverReadChunk(fp,len)
FILE *fp;
ULONG len;
{
 ULONG i;
```

```
 BYTE dummy;
 for(i=0;i<len;i++)
   dummy=GetByte(fp);
 if(len&1L)
   dummy=GetByte(fp);
}
VOID LookForPic(fp)
FILE *fp;
{
 BOOL body;
 LONG length,id,i;
 USHORT Mode;
 BYTE dummy;
 body=FALSE;
 length=GetLong(fp);
 id=GetLong(fp);
 if(id!=ID_ILBM)
  Quit("Geen IFF file, ILBM
    error\n");
 while(!(body))
 {
   id=GetLong(fp);
   length=GetLong(fp);
   switch(id)
      {
      case(ID_BMHD):
         loc_bmhd=(UBYTE
*)(&MyBmhd);
         for(i=0;i<length;i++)
            *loc_bmhd++ =
GetByte(fp);
         if(length&1)
            dummy=GetByte(fp);

    Scherm.LeftEdge=MyBmhd.y;

    Scherm.TopEdge=MyBmhd.x;

    Scherm.Width=MyBmhd.Width;

    Scherm.Height=MyBmhd.Height
    ;

    Scherm.Depth=MyBmhd.Depth;
         Scherm.DetailPen=0;
         Scherm.BlockPen=0;
         Mode=NULL;
         if(MyBmhd.Width>400)
    Mode=Mode|HIRES;

    if(MyBmhd.Height>256)
    Mode=Mode|LACE;
         if(MyBmhd.Depth==6
    && MyBmhd.Width<400)
    Mode=Mode|HAM;

    Scherm.ViewModes=Mode;

    Scherm.Type=CUSTOMSCREEN;
         Scherm.Font=NULL;

    Scherm.DefaultTitle=(UBYTE
    *)"Commodore INFO
    LoadILBM";
         Scherm.Gadgets=NULL;

    Scherm.CustomBitMap=NULL;
         if(!(PScreen=(struct
    Screen *)

    OpenScreen(&Scherm)))
                   Quit("Te
    weinig geheugen voor ILBM
    scherm\n");
         break;

      case(ID_CMAP):
         loc_cmap=kleuren;
         for(i=0;i<length;i++)
            *loc_cmap++ =
    GetByte(fp);
         colors=length;
         if(length&1L)
           dummy=GetByte(fp);
         ComputeCMap();
         break;

      case(ID_BODY):
         ReadBody(fp);
         body=TRUE;
         break;

      default:

    OverReadChunk(fp,length);
        break;
      }
    }
}
ComputeCMap()
{
 LONG i,c,red,green,blue;
 loc_cmap=kleuren;
 c=colors/3L;
 for(i=0;i<c;i++)
   {
    red=(LONG)(*loc_cmap++ >>
   4);
    green=(LONG)(*loc_cmap++
   >> 4);
    blue=(LONG)(*loc_cmap++
   >> 4);

   SetRGB4(&PScreen->ViewPort,
   i,red,green,blue);
   }
}

UBYTE *UnPackRow(bt,l,fp)
UBYTE *bt;
UWORD l;
FILE *fp;
{
 BYTE n;
 UWORD i,j,k,count;
 count=0;

 if(MyBmhd.Compression==1)
   {
    while(count<l)
      {
       n=GetByte(fp);
       if(n<0)
         {
          if(n>-128)
            {
             k=GetByte(fp);

   for(j=0;j<((-n)+1);j++)
                {
                 *bt++ = k;
                 count++;
                }
            }
         }
      if(n>=0)
        {
         for(j=0;j<(n+1);j++)
           {
            *bt++=GetByte(fp);
            count++;
           }
        }
   }
 }
if(MyBmhd.Compression==0)
 {
  for(j=0;j<l;j++)
    *bt++=GetByte(fp);
 }
 return(bt);
}

ReadBody(fp)
FILE *fp;
{
 UWORD i,j,x,y;
 UBYTE *maps[7],*dummy;
 x=(MyBmhd.Width>>3);
 y=MyBmhd.Height;
 for(j=0;j<MyBmhd.Depth;j++)

    maps[j]=PScreen->BitMap.Pla
    nes[j];

 for(i=0;i<y;i++)
   {
    for(j=0;j<MyBmhd.Depth;j++)

    maps[j]=UnPackRow(maps[j],x
    ,fp);
    if(MyBmhd.Masking==1 ||
    MyBmhd.Masking==3)

    dummy=UnPackRow(kleuren,x,f
    p);
   }
}

main(argc,argv)
SHORT argc;
char *argv[];
{
 int i;
 char *MouseL=(char
    *)0xbfe001;

 if(!(IntuitionBase=(struct
    IntuitionBase *)

    OpenLibrary("intuition.libr
    ary",0L)))
        Quit("Geen
    intuition.library\n");
 if(!(GfxBase=(struct GfxBase
    *)

    OpenLibrary("graphics.libra
    ry",0L)))
        Quit("Geen
    graphics.library\n");

 if(argc!=2)
   Quit("Syntax van LoadILBM:
    LoadILBM filenaam\n");
 LoadPicture(argv[1]);

 while((*MouseL &
    0x40)==0x40);

 CloseScreen(PScreen);
 CloseLibrary(GfxBase);
 CloseLibrary(IntuitionBase);
}
```